CHAPTER 3

Loading System Images and Configuration Files

This chapter describes how to load system images and configuration files. The system images contain the system software, and the configuration files contain commands entered to customize the function of the communication server. The instructions in this chapter describe how to copy system images from communication servers to network servers (and vice versa), display and compare different configuration files, and list the system software version running on the communication server.

This chapter also describes the AutoInstall procedure, which you can use to automatically configure and enable a new communication server upon startup.

For a complete description of the commands mentioned in this chapter, refer to the "System Image and Configuration File Load Commands" chapter in the *Access and Communication Servers Command Reference* publication.

Note You also can use the **setup** command and its interactive prompts to create a basic configuration file. See the *Access and Communication Servers Getting Started Guide* for more information.

One of the first attempts to use the network as a resource in the UNIX community resulted in the design and implementation of the remote copy protocol (rcp), the remote shell (rsh) protocol, and their commands. The rsh and rcp protocols give users the ability to execute commands remotely and copy files to and from a file system residing on a remote host or server on the network.

From the communication server, you can use rsh to execute commands on remote systems to which you have access. When you issue the rsh command, a shell is started on the remote system. The shell allows you to execute commands on the remote system without having to log into the system.

In other words, you do not need to connect to the system or communication server and then disconnect after you execute a command if you use rsh. For example, you can use rsh to remotely look at the status of other communication servers without connecting to the communication server, executing the command, and then disconnecting from the communication server. This is useful for looking at statistics on many different communication servers.

To gain access to a remote system running rsh, such as a UNIX host, you must be configured in the system's *.rhosts* file or its equivalent. On UNIX systems, the *.rhosts* file identifies trusted users who can remotely execute commands on the system.

You can enable rsh support on a communication server to allow users on remote systems to execute commands on the communication server. However, our implementation of rsh does not support an *.rhosts* file. Instead, you configure a local authentication database to control access to the communication server by users attempting to execute commands remotely using rsh. A local

authentication database is similar in concept and use to a UNIX *.rhosts* file. Each entry that you configure in the authentication database identifies the local user, the remote host, and the remote user.

The rcp copy commands rely on the rsh server (or daemon) on the remote system. To copy files using rcp, you do not need to create a server for file distribution, as you do with TFTP. You only need to have access to a server that supports the remote shell (rsh). (Most UNIX systems support rsh.) Because you are copying a file from one place to another, you must have read permission on the source file and write permission on the destination file. If the destination file does not exist, rcp creates it for you.

Although our rcp implementation emulates the behavior of the UNIX rcp implementation—copying files among systems on the network—our command syntax differs from the UNIX rcp command syntax. Our rcp support offers a set of copy commands that use rcp as the transport mechanism. These rcp copy commands are similar in style to our TFTP copy commands, but they offer an alternative that provides faster performance and reliable delivery of data. This is because the rcp transport mechanism is built on and uses the Transmission Control Protocol/Internet Protocol (TCP/IP) stack, which is connection-oriented. You can use the Cisco rcp commands to copy system images and configuration files from the communication server to a network server and vice versa.

You can also enable rcp support on the communication server to allow users on remote systems to copy files to and from the communication server.

System Image and Configuration File Load Task List

You can perform the tasks in the following sections to load system images and configuration files.

- Use the AutoInstall Procedure
- Enter Global Configuration Mode
- Modify the Configuration Register Boot Field
- Specify the System Image the Communication Server Loads upon Restart
- Specify the Configuration File the Communication Server Loads upon Restart
- Change the Buffer Size for Loading Configuration Files
- Compress Configuration Files
- Manually Load a System Image
- Configure a Communication Server as a TFTP Server
- Configure a Communication Server to Support Incoming rcp Requests and rsh Commands
- Configure a Communication Server as a RARP Server
- Configure the Remote Username for rcp Requests
- Specify SLIP Extended BOOTP Requests
- Specify MOP Server Boot Requests
- Copy System Images from a Network Server to Flash Memory Using TFTP
- Copy System Images from a Network Server to Flash Memory Using rcp
- Use Flash Load Helper
- Verify the Image in Flash Memory
- Use Dual Flash Bank

- Copy System Images from Flash Memory to a Network Server Using TFTP
- Copy System Images from Flash Memory to a Network Server Using rcp
- Copy a Configuration File from a Network Server to the Communication Server Using rcp
- Copy a Configuration File from the Communication Server to a Network Server
- Copy a Configuration File from the Communication Server to a Network Server Using rcp
- Display System Image and Configuration Information
- Clear the Contents of Nonvolatile Memory
- Reexecute the Configuration Commands in Nonvolatile Memory
- Remotely Execute Commands Using rsh
- Use Flash Memory as a TFTP Server

Use the AutoInstall Procedure

This section provides information about AutoInstall, a procedure that enables you to configure a new communication server automatically and dynamically. The AutoInstall procedure involves connecting a new communication server to a network on which there is an existing preconfigured communication server, turning on the new communication server, and having it immediately enabled with a configuration file that is automatically downloaded from a Trivial File Transfer Protocol (TFTP) server.

The following sections provide the requirements for AutoInstall and present an overview of how the procedure works. To start the procedure, go to “Perform the AutoInstall Procedure” later in this section.

AutoInstall Requirements

For the AutoInstall procedure to work, your system must meet the following requirements:

- The existing preconfigured communication server must be running Software Release 8.3 or later.
- The new communication server must be running Software Release 9.1 or later.
- Both communication servers must be physically attached to the network by means of one or more of the following interface types: Ethernet, Token Ring, or serial with HDLC encapsulation (the default encapsulation).
- You must complete procedures 1 and either 2 or 3:
  - Procedure 1: A configuration file for the new communication server must reside on a TFTP server. This file can contain the new communication server’s full configuration or the minimum needed for the administrator to Telnet into the new communication server for configuration.
  - Procedure 2: A file named *network-confg* also must reside on the server. The file must have an Internet Protocol (IP) host name entry for the new communication server. The server must be reachable from the existing communication server.
  - Procedure 3: An IP address-to-host name mapping for the new communication server must be added to a Domain Name System (DNS) database file.

- If the existing communication server is to help automatically install the new communication server via a High-Level Data Link Control (HDLC)-encapsulated serial interface using Serial Line Address Resolution Protocol (SLARP), that interface must be configured with an IP address whose host portion has the value 1 or 2. Subnet masks of any size are supported.
- If the existing communication server is to help autoinstall the new communication server via an Ethernet or Token Ring interface using BOOTP or Reverse Address Resolution Protocol (RARP), a BOOTP or RARP server also must be set up to map the new communication server's Media Access Control (MAC) address to its IP address.
- IP helper addresses might need to be configured in order to forward the TFTP and DNS broadcast requests from the new communication server to the host that is providing those services.

How AutoInstall Works

Once the requirements for using AutoInstall are met, the dynamic configuration of the new communication server occurs in the following order:

1. The new communication server acquires its IP address.
2. Depending upon the interface connection between the two communication servers, the new communication server's IP address is dynamically resolved by either SLARP, BOOTP, or RARP requests.
3. The new communication server resolves its IP address-to-host name mapping.
4. The new communication server automatically requests and downloads its configuration file from a TFTP server.

Acquiring the New Communication Server's IP Address

The new communication server (*newcommserver*) resolves its interface's IP addresses by one of the following means:

- If *newcommserver* is connected by an HDLC-encapsulated serial line to the existing communication server (*existing*), *newcommserver* sends a SLARP request to *existing*.
- If *newcommserver* is connected to an Ethernet or Token Ring interface, it broadcasts BOOTP and RARP requests.

The existing communication server (*existing*) responds in one of the following ways depending upon the request type:

- In response to a SLARP request, *existing* sends a SLARP reply packet to *newcommserver*. The reply packet contains the IP address and netmask of *existing*. If the host portion of the IP address in the SLARP response is 1, *newcommserver* will configure its interface using the value 2 as the host portion of its IP address and vice versa. (See Figure 3-1.)

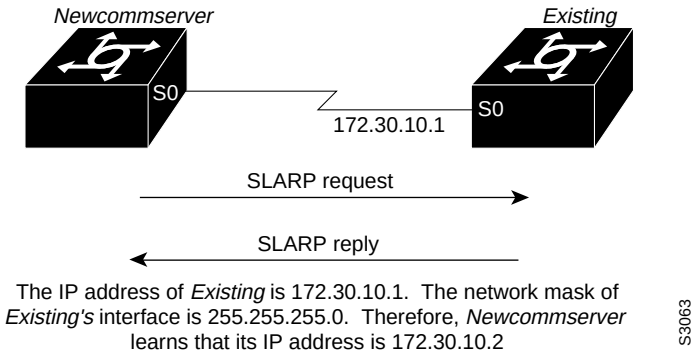

Figure 3-1 Using SLARP to Acquire the New Communication Server's IP Address

- In response to BOOTP/RARP requests, an IP address is sent from the BOOTP or RARP server to *newcommserver.*

  A BOOTP or RARP server must have already been set up to map the *newcommserver's* MAC address to its IP address. If the BOOTP server does not reside on the directly attached network segment, communication servers between *newcommserver* and the BOOTP server can be configured using the **ip helper-address** command to allow the request and response to be forwarded between segments, as shown in Figure 3-2.

Figure 3-2 Using BOOTP/RARP to Acquire the New Communication Server's IP Address

As of Software Release 9.21, communication servers can be configured to act as RARP servers.

As soon as one interface resolves its IP address, the communication server will move on to resolve its host name. Therefore, only one IP address needs to be set up using either SLARP, BOOTP, or RARP.

Resolving the IP Address to the Host Name

The new communication server resolves its IP address-to-host name mapping by sending a TFTP broadcast requesting the file network-confg, as shown in Figure 3-3.

The *network-confg* file is a configuration file generally shared by several communication servers. In this case, it is used to map the IP address the new communication server just obtained dynamically to the name of the new communication server. The file *network-confg* must reside on a reachable TFTP server and must be globally readable.

The following is an example of a minimal network-confg file that maps the IP address of the new communication server (131.108.10.2) to the name *newcommserver*. The address of the new communication server was learned via SLARP and is based on *existing's* IP address of 131.108.10.1.

```
ip host newcommserver 131.108.10.2
```

If *newcommserver* does not receive a *network-confg* file, or if the IP address-to-host name mapping does not match the newly acquired IP address, *newcommserver* sends a DNS broadcast. If DNS is configured and has an entry that maps *newcommserver*'s SLARP, BOOTP, or RARP-acquired IP address to its name, *newcommserver* successfully resolves its name.

If DNS does not have an entry mapping *newcommserver*'s SLARP, BOOTP, or RARP-acquired address to its name, the new communication server cannot resolve its host name. The new communication server attempts to download a default configuration file as described in the next section, and failing that, enters setup mode.

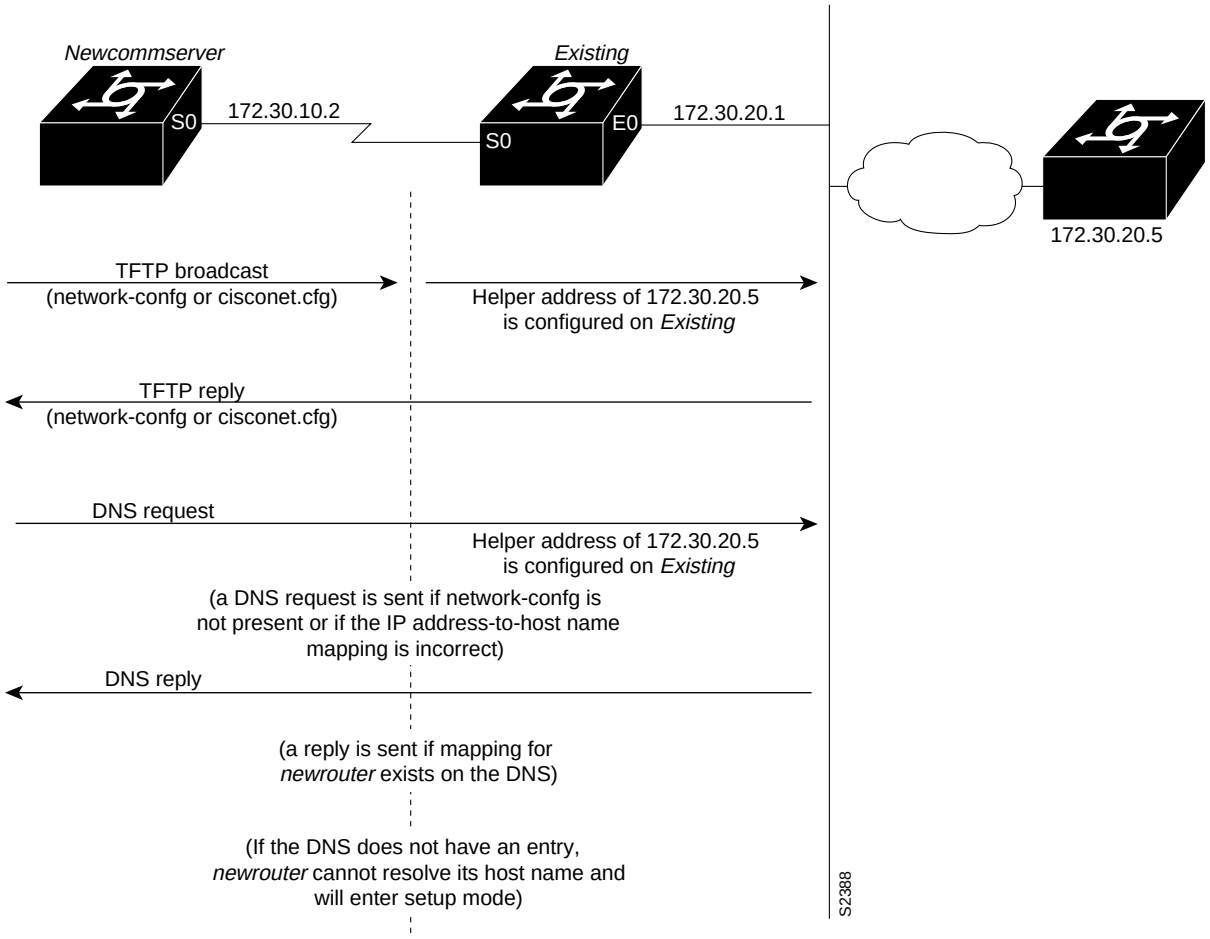

Figure 3-3 Dynamically Resolving the IP Address-to-Host Name Mapping

Downloading the New Communication Server's Host Configuration File

After the communication server successfully resolves its host name, *newcommserver* sends a TFTP broadcast requesting the file *newcommserver-confg*. The name *newcommserver-confg* must be in all lowercase, even if the true host name is not. If *newcommserver* cannot resolve its host name, it sends a TFTP broadcast requesting the default host configuration file *communicationserver-confg*. The file is downloaded to *newcommserver* where the configuration commands take effect immediately.

If the host configuration file contains only the minimal information, the administrator uses Telnet to get into *existing*, from there using Telnet to *newcommserver*, and then run the **setup** command to configure *newcommserver*. Refer to the *Access and Communication Servers Getting Started Guide* for details on the **setup** command.

If the host configuration file is complete, *newcommserver* should be fully operational. You can enter the **enable** command (with the system administrator password) at the system prompt on *newcommserver*, and then issue the **write memory** command to save the information in the recently obtained configuration file into nonvolatile memory. If a reload occurs, *newcommserver* simply loads its configuration file from nonvolatile memory.

If the TFTP request fails, or if *newcommserver* still has not obtained the IP addresses of all its interfaces, and those addresses are not contained in the host configuration file, then *newcommserver* enters setup mode automatically. Setup mode prompts for manual configuration of the communication server via the console. The new communication server continues to issue broadcasts to attempt to learn its host name and obtain any unresolved interface addresses. The broadcast frequency will dwindle to every ten minutes after several attempts. Refer to the *Access and Communication Servers Getting Started Guide* for details on the **setup** command.

Perform the AutoInstall Procedure

To dynamically configure a new communication server using AutoInstall, complete the following tasks. Steps 1, 2, and 3 are completed by the central administrator. Step 4 is completed by the person at the remote site.

Step 1 Modify the existing communication server's configuration to support the AutoInstall procedure.

Step 2 Set up the TFTP server to support the AutoInstall procedure.

Step 3 Set up BOOTP or RARP server if needed (required for AutoInstall using an Ethernet or Token Ring interface; not required for AutoInstall using an HDLC-encapsulated serial interface).

Step 4 Connect the new communication server to the network.

Modify the Existing Communication Server's Configuration

You can use either of the following types of interface:

- An HDLC-encapsulated serial line, the default configuration for a serial line
- An Ethernet or Token Ring interface

Use a Serial Interface (HDLC Encapsulation) Connection

To set up AutoInstall via a serial line with HDLC encapsulation (the default), complete the following tasks to configure the existing communication server:

Task		Command
Step 1	Enter configuration mode from privileged EXEC mode.	**configure terminal**
Step 2	Configure the serial interface that connects to the new communication server with HDLC encapsulation (the default).	**interface serial** *interface-number*[1]
Step 3	Enter an IP address for the interface The host portion of the address must have a value of 1 or 2.	**ip address** *address mask*[2]
Step 4	Configure a helper address for the serial interface to forward broadcasts associated with the TFTP, BOOTP, and DNS requests.	**ip helper-address** *address*[2]
Step 5	Optionally, configure a DCE clock rate for the serial line, unless an external clock is being used. This step is needed only for DCE appliques.	**clockrate** *bits per second*[1]
Step 6	Exit configuration mode.	**Ctrl-Z**
Step 7	Save the configuration changes to nonvolatile memory.	**write memory**

1. These commands are documented in the "Interface Configuration Commands" chapter in the *Access and Communication Servers Command Reference* publication.
2. These commands are documented in the "IP Commands" chapter in the *Access and Communication Servers Command Reference* publication.

You must use a DTE interface on the new communication server because there is no default clock rate for a DCE interface.

In the following example, the existing communication server's configuration file contains the commands needed to configure the communication server for AutoInstall on a serial line:

```
cs1# configure terminal
cs1(config)# interface serial 0
cs1(config)# ip address 131.108.10.1 255.255.255.0
cs1(config)# ip helper-address 131.108.20.5
Ctrl-Z
cs1# write memory
```

Use an Ethernet or Token Ring Interface Connection

To set up AutoInstall using an Ethernet or Token Ring interface, complete the following tasks as needed to modify the configuration of the existing communication server. Typically, the local area network (LAN) interface and IP address are already configured on the existing communication server. You might need to configure an IP helper address if the TFTP server is not on the same network as the new communication server.

Task		Command
Step 1	Enter configuration mode from privileged EXEC mode.	**configure terminal**
Step 2	Configure a LAN interface.	**interface** {**ethernet** | **tokenring**} *interface-number*[1]
Step 3	Enter an IP address for the interface.	**ip address** *address mask*[2]

Task		Command
Step 4	Optionally, configure a helper address to forward broadcasts associated with the TFTP, BOOTP, and DNS requests.	**ip helper-address** *address*[2]
Step 5	Exit configuration mode.	**Ctrl-Z**
Step 6	Save the configuration changes to nonvolatile memory.	**write memory**

1. This command is documented in the "Interface Configuration Commands" chapter in the *Access and Communication Servers Command Reference* publication.
2. These commands are documented in the "IP Commands" chapter in the *Access and Communication Servers Command Reference* publication.

In the following example, the existing communication server's configuration file contains the commands needed to configure the communication server for AutoInstall on an Ethernet interface:

```
cs1# configure terminal
cs1(config)# interface Ethernet 0
cs1(config-if)# ip address 131.108.10.1 255.255.255.0
cs1(config-if)# ip helper-address 131.108.20.5
Ctrl-Z
cs1# write memory
```

Set up the TFTP Server

For AutoInstall to work correctly, the new communication server must be able to resolve its host name and then download a *name-confg* file from a TFTP server. The new communication server can resolve its host name by using a *network-confg* file downloaded from a TFTP server or by using the DNS.

To set up a TFTP server to support AutoInstall, complete the following tasks. Step 2 includes two ways to resolve the new communication server's host name:

Task		Command
Step 1	Enable TFTP on a server.	Consult your host vendor's TFTP Server documentation and RFCs 906 and 783.
Step 2	If you want to use a *network-confg* file to resolve the new communication server's name, create the file *network-confg* containing an IP address-to-host name mapping for the new communication server. Enter the **ip host** command into the TFTP config file, not into the communication server. The IP address must match the IP address that is to be dynamically obtained by the new communication server.	**ip host** *hostname address*[1]
	or	
	If you want to use the DNS to resolve the new communication server's name, create an address-to-name mapping entry for the new communication server in the DNS database. The IP address must match the IP address that is to be dynamically obtained by the new communication server.	Contact the DNS administrator or refer to RFCs 1101 and 1183.

Task		Command
Step 3	Create the file *name-confg*, which should reside in the *tftpboot* directory on the TFTP server. The *name* part of *name-confg* must match the host name you assigned for the new communication server in the previous step. Enter into this file configuration commands for the new communication server.	See the appropriate chapter in this guide for specific commands.

1. This command is documented in the "IP Commands" chapter in the *Access and Communication Servers Command Reference* publication.

The *name-confg* file can contain either the new communication server's full configuration or a minimal configuration.

The minimal configuration file consists of a virtual terminal password and an enable password. It allows an administrator to Telnet into the new communication server to configure it. If you are using BOOTP or RARP to resolve the address of the new communication server, the minimal configuration file must also include the IP address to be obtained dynamically using BOOTP or RARP.

You can use the **write network** command to help you generate the configuration file that you will download during the Autoinstall process.

Note The existing communication server might need to forward TFTP requests and response packets if the TFTP server is not on the same network segment as the new communication server. When you modified the existing communication server's configuration, you specified an IP helper address for this purpose.

You can save a minimal configuration under a generic *newcommserver-confg* file. Use the **ip host** command in the *network.confg* file to specify *newcommserver* as the host name with the address you will be dynamically resolving. The new communication server should then resolve its IP address, host name and minimal configuration automatically. Use Telnet to connect to the new communication server from the existing communication server and use the **setup** facility to configure the rest of the interfaces. For example, the line in the *network-confg* file could be similar to the following:

```
ip host newcommserver 131.108.170.1
```

The following host configuration file contains the minimal set of commands needed for AutoInstall using SLARP or BOOTP:

```
enable-password letmein
!
line vty 0
password letmein
!
end
```

The preceding example shows a minimal configuration for connecting from a communication server one hop away. From this configuration, use the **setup** facility to configure the rest of the interfaces. If the communication server is more than one hop away, you also must include routing information in the minimal configuration.

The following minimal network configuration file maps the new communication server's IP address, 131.108.10.2, to the host name *newcommserver*. The new communication server's address was learned via SLARP and is based on the *existing* communication server's IP address of 131.108.10.1.

```
ip host newcommserver 131.108.10.2
Set up the BOOTP or RARP Server
```

If the new communication server is connected to the existing communication server using an Ethernet or Token Ring interface, you must configure a BOOTP or RARP server to map the new communication server's MAC address to its IP address. If the new communication server is connected to the existing communication server using a serial line with HDLC encapsulation, the steps in this section are not required.

To configure a BOOTP or RARP server, complete one of the following tasks:

Task	Command
If BOOTP is to be used to resolve the new communication server's IP address, configure your BOOTP server.	Refer to your host vendor's manual pages and to RFCs 951 and 1395
If RARP is to be used to resolve the new communication server's IP address, configure your RARP server.	Refer to your host vendor's manual pages and to RFC 903

Note If the RARP server is not on the same subnet as the new communication server, use the **ip rarp-server** command to configure the existing communication server to act as a RARP server. See the section "Configure a Communication Server as a RARP Server" later in this chapter.

The following host configuration file contains the minimal set of commands needed for AutoInstall using RARP. It includes the IP address that will be obtained dynamically via BOOTP or RARP during the AutoInstall process. When RARP is used, this extra information is needed to specify the proper netmask for the interface.

```
interface ethernet 0
ip address 131.108.10.2 255.255.255.0
enable-password letmein
!
line vty 0
password letmein
!
end
```

Connect the New Communication Server to the Network

Connect the new communication server to the network using either an HDLC-encapsulated serial interface or an Ethernet or Token Ring interface. After the communication server successfully resolves its host name, *newcommserver* sends a TFTP broadcast requesting the file *name-confg*. The communication server name must be in all lowercase, even if the true host name is not. The file is downloaded to the new communication server where the configuration commands take effect

immediately. If the configuration file is complete, the new communication server should be fully operational. To save the complete configuration to nonvolatile memory, complete the following steps:

Task		Command
Step 1	Enter privileged mode at the system prompt on the new communication server.	**enable** [1] *password*
Step 2	Save the information from the *name-config* file into nonvolatile memory.	**write memory**

1. This command is documented in the "User Interface Commands" chapter in the *Access and Communication Servers Command Reference* publication.

Caution Verify that the existing and new communication servers are connected before entering the **write memory** EXEC command to save these configuration changes. Use the **ping** EXEC command to verify connectivity. If an incorrect configuration file is downloaded, the new communication server will load nonvolatile memory configuration information before it can enter AutoInstall mode.

If the configuration file is a minimal configuration file, the new communication server comes up, but with only one interface operational. Complete the following steps to connect to the new communication server and configure it:

Task		Command
Step 1	Establish a Telnet connection to the existing communication server.	**telnet** *existing*[1]
Step 2	From the existing communication server, establish a Telnet connection to the new communication server.	**telnet** *newcommserver*[1]
Step 3	Enter privileged EXEC mode.	**enable**[2] *password*
Step 4	Enter **setup** mode to configure the new communication server.	**setup**[3]

1. These commands are documented in the "Terminal Service Connections" chapter in the *Cisco Access Connection Guide* publication.
2. This command is documented in the "User Interface Commands" chapter in the *Access and Communication Servers Command Reference*.
3. This command is documented in the *Access and Communication Servers Getting Started Guide*.

Enter Global Configuration Mode

To enter global configuration mode, enter the EXEC command **configure** at the privileged-level EXEC prompt. The communication server responds with the following prompt asking you to specify the terminal, nonvolatile memory, or a file stored on a network server as the source of configuration commands:

```
Configuring from terminal, memory, or network [terminal]?
```

These three methods are described in the next three sections.

The communication server accepts one configuration command per line. You can enter as many configuration commands as you want.

You can add comments to a configuration file describing the commands you have entered. Precede a comment with an exclamation point (!). Comments are *not* stored in nonvolatile memory or in the active copy of the configuration file. In other words, comments do not show up when you list the active configuration with the **write terminal** EXEC command or list the configuration in nonvolatile memory with the **show configuration** EXEC command. Comments are stripped out of the configuration file when it is loaded to the communication server. However, you can list the comments in configuration files stored on a TFTP or MOP server.

Configure the Communication Server from the Terminal

To configure the communication server from the terminal, complete the following tasks:

Task		Command
Step 1	From privileged EXEC mode, enter global configuration mode and select the terminal option.	**configure terminal**
Step 2	Enter the necessary configuration commands.	See the appropriate chapter for specific configuration commands.
Step 3	Exit global configuration mode.	**Ctrl-Z**
Step 4	Save the configuration file modifications to nonvolatile memory.	**write memory**

In the following example, the communication server is configured from the terminal. The comment "The following command provides the communication server host name" identifies the purpose of the next command line. The **hostname** command changes the communication server name from cs1 to cs2. By pressing Ctrl-Z, the user quits configuration mode. The command **write memory** loads the configuration changes into nonvolatile memory.

```
cs1# configure terminal
cs1(config)# !The following command provides the communication server host name.
cs1(config)# hostname cs2
Ctrl-Z
cs2# write memory
```

Nonvolatile memory stores the current configuration information in text format as configuration commands, recording only nondefault settings. The memory is checksummed to guard against corrupted data.

As part of its startup sequence, the communication server startup software always checks for configuration information in nonvolatile memory. If nonvolatile memory holds valid configuration commands, the communication server executes the commands automatically at startup. If the communication server detects a problem with the nonvolatile memory or the configuration it contains, it enters **setup** mode and prompts for configuration. Problems can include a bad checksum for the information in nonvolatile memory or the absence of critical configuration information. See the publication *Troubleshooting Internetworking Systems* for troubleshooting procedures. See the *Access and Communication Servers Getting Started Guide* for details on setup information.

Configure the Communication Server from Nonvolatile Memory

You can configure the communication server from nonvolatile memory by reexecuting the configuration commands stored in nonvolatile memory. To do so, complete the following task in privileged EXEC mode:

Task	Command
Configure the communication server from nonvolatile memory.	**configure memory**

Configure the Communication Server from a File on a Remote Host

You can configure the communication server by retrieving and adding to the configuration file stored on one of your network servers. To do so, complete the following tasks:

Task		Command
Step 1	Enter global configuration mode with the network option.	**configure network**
Step 2	At the system prompt, select a host or network configuration file. The network configuration file contains commands that apply to all network servers and terminal servers on the network. The host configuration file contains commands that apply to one network server in particular.	**host** or **network**
Step 3	At the system prompt, enter the optional IP address of the remote host from which you are retrieving the configuration file.	*ip-address*
Step 4	At the system prompt, enter the name of the configuration file or accept the default name.	*filename*
Step 5	Confirm the configuration filename that the system supplies.	**y**

In the following example, the communication server is configured from the file *tokyo-config* at IP address 131.108.2.155:

```
cs1# configure network
Host or network configuration file [host]?
IP address of remote host [255.255.255.255]? 131.108.2.155
Name of configuration file [tokyo-confg]?
Configure using tokyo-confg from 131.108.2.155? [confirm] y
Booting tokyo-confg from 131.108.2.155:!! [OK - 874/16000 bytes]
```

Copy a Configuration File to NVRAM

To load a configuration file directly into NVRAM without affecting the running configuration, perform the following task in privileged EXEC mode:

Task	Command
Load a configuration file directly into NVRAM without affecting the running configuration.	**configure overwrite**

Modify the Configuration Register Boot Field

The order in which the communication server looks for configuration information depends upon the boot field setting in the configuration register. The configuration register is a 16-bit register. The lowest four bits of the configuration register (bits 3, 2, 1, and 0) form the boot field. Depending upon the value that you set in the boot field of the configuration register, the system is configured to boot according to one of the following methods:

- Manual boot, initiated by using the **b** command at the ROM monitor prompt
- Automatic boot from read-only memory
- Automatic boot using the **boot system** commands in nonvolatile memory

Table 3-1 lists the configuration register settings for these three boot methods.

Table 3-1 Configuration Register Boot Values

Boot Value	Description
0x100	Sets the boot field to binary 0000, which configures the system to boot manually; initiated by using the **b** command at the ROM monitor prompt.
0x101	Sets the boot field to binary 0001, which configures the system to boot automatically, using configuration information stored in read-only memory.
0x102 to 0x10F	Set the boot field to binary 0010 to 0111, which configures the system to boot automatically using the **boot system** commands in nonvolatile memory. (These values set the boot field to binary 0010-111.) If there are no **boot system** commands in nonvolatile memory, the system uses the configuration register value to form a filename from which to netboot a default system image stored on a network server. (Refer to the appropriate hardware guide for details on default filenames.)

The value you enter is stored in nonvolatile memory, but does not take effect until you reboot the communication server.

For communication servers running Software Release 9.1 or later, the configuration register can only be changed on the processor card or through DIP switches located on the communication server. Refer to the appropriate hardware installation guide for details.

For the ASM-CS, 500-CS, and Cisco 2500 series running Software Release 9.1 or later, you can change the configuration register by completing the following tasks:

Task		Command
Step 1	Enter global configuration mode and select the terminal option.	**configure terminal**
Step 2	Modify the default configuration register setting.	**config-register** *value*[1]
Step 3	From privileged EXEC mode, list the current configuration register setting and the new configuration register setting, if any, that will be used the next time the communication server is reloaded. or From the ROM monitor prompt, list the value of the boot field in the configuration register.	**show version** **o**
Step 4	Exit global configuration mode.	**Ctrl-Z**

1. This command works only if you have IOS Release 10.2 boot ROMs. Systems using older boot ROMs (9.1, 9.14, or 9.21) still must use the hardware configuration register.

In the following example, the configuration register is set so that the communication server will boot automatically from the Flash memory default file. The last line of the output of the **show version** command indicates that a new configuration register setting (0x10F) will be used the next time the communication server is reloaded.

```
cs1# configure terminal
cs1(config)# config-register 0x10F
Ctrl-Z
cs1# show version
GS Software, Version 9.0(1)
Copyright (c) 1986-1994 by cisco Systems, Inc.
Compiled Mon 27-Jun-94 11:04

ROM: System Bootstrap, Version (3.3), SOFTWARE
cs1 uptime is 2 days, 10 hours, 0 minutes
System restarted by reload
System image file is unknown, booted via tftp from 131.108.13.111
Host configuration file is "thor-boots", booted via tftp from 131.108.13.111
Network configuration file is "network-confg", booted via tftp from
131.108.13.111

Cisco 2500 (68030) processor with 1024K/1024K bytes of memory.
Processor board serial number 01234567.
X.25 software, Version 2.0, NET2, BFE and GOSIP compliant.
Bridging software.
SuperLAT software (copyright 1990 by Meridian Technology Corp).
Authorized for Enterprise software set. (0x0)
1 Ethernet/IEEE 802.3 interface.
2 Serial network interfaces.
8 terminal lines.
32K bytes of non-volatile configuration memory.
4096K bytes of processor board system flash. (Read only mode)
Configuration register is 0x0
```

Specify the System Image the Communication Server Loads upon Restart

You can enter multiple boot commands in nonvolatile memory configuration to provide backup methods for loading a system image onto the communication server. There are three ways to load a system image:

- From Flash memory—Flash allows you to copy new system images without changing erasable programmable read-only memory (EPROM). Information stored in Flash is not vulnerable to network failures that might occur when loading system images from servers.
- From a network server—In case Flash memory becomes corrupted, specifying a system image to be loaded from a network server using TFTP, rcp, or MOP provides a backup boot method for the communication server.
- From ROM—In case of both network failure and Flash memory corruption, specifying a system image to be loaded from ROM provides a final backup boot method. System images stored in ROM may not always be as complete as those stored in Flash memory or on network servers.

You can enter the different types of boot commands in any order in nonvolatile memory configuration. If you enter multiple boot commands, the communication server tries them in the order they are entered.

Load from Flash Memory

Flash memory is available for the ASM-CS. With a CSC-MC+ Flash memory card and CSC-MCI controller and appropriate cables, system software images can be written to Flash memory for booting. Depending on the hardware platform, Flash memory might be available as EPROMs, single in-line memory modules (SIMMs), or memory cards. Check the appropriate hardware installation and maintenance guide for information about types of Flash memory available on a specific platform.

Software images can be stored, booted, and rewritten into Flash memory as necessary. Flash memory can reduce the effects of network failure by reducing dependency on files that can only be accessed over the network.

Flash memory allows you to do the following:

- Copy the system image to Flash memory using TFTP.
- Copy the system image to Flash memory using rcp.
- Boot a communication server from Flash memory either automatically or manually.
- Copy the Flash memory image to a network server using TFTP or rcp.

Note Use of Flash memory is subject to the terms and conditions of the software license agreement that accompanies your product.

Flash memory features include the following:

- It can be remotely loaded with multiple system software images through TFTP or rcp transfers (one transfer for each file loaded).
- It allows a communication server to be booted manually or automatically from a system software image stored in Flash memory. Booting directly from ROM or booting from a network server using TFTP or rcp are also available options.
- Flash memory provides write protection against accidental erasing or reprogramming.

Note Booting from ROM is faster than booting from Flash. However, if you are booting from a network server, Flash is faster and more reliable.

Security Precautions

Take the following precautions when loading from Flash memory:

- Flash memory provides write protection against accidental erasing or reprogramming.
- The system image stored in Flash memory can be changed only from a privileged EXEC command session on the console terminal.

Flash Memory Configuration

The following list is an overview of how to configure your system to boot from Flash memory. It is not a step-by-step set of instructions; rather, it is an overview of the process of using the Flash capability.

Refer to the appropriate Cisco hardware installation and maintenance publication for complete instructions for installing the hardware and for information about the jumper settings required for your configuration.

1. Set your system to boot from ROM software.

   The configuration register boot filed value might need to be changed.

2. Restore the system configuration, if necessary.
3. Copy the system image to Flash memory using rcp or TFTP.
4. Configure from the terminal to automatically boot from the desired file in Flash memory.
5. Set your system to boot from a file in Flash memory.

   The configuration register boot field value might need to be changed.

6. Turn the system power off, then on again and reboot your system to ensure that all is working as expected.

Once you have successfully configured Flash memory, you might want to configure the system with the **no boot system flash** command to revert back to booting from ROM.

To configure the communication server to boot automatically from an image in Flash memory by completing the following tasks:

Task		Command
Step 1	Enter global configuration mode from the terminal.	**configure terminal**
Step 2	Enter the filename of an image stored in Flash memory.	**boot system flash** [*filename*]
Step 3	Set the configuration register to enable loading of the system image from Flash memory.	**config-register** *value*
Step 4	Exit global configuration mode.	**Ctrl-Z**
Step 5	Save the configuration information to nonvolatile memory.	**write memory**

Automatically booting from Flash memory requires changing the processor's configuration register. See the section entitled "Modifying the Configuration Register Boot Field" earlier in this chapter. Use the **show version** command to list the current configuration register setting.

If you enter more than one image filename, the communication server tries them in the order entered.

If a filename already appears in the configuration file and you want to specify a new filename, remove the existing filename with the **no boot system flash** *filename* command.

Note The **no boot system** configuration command disables all **boot system** configuration commands regardless of argument. Specifying the **flash** keyword or the *filename* argument with the **no boot system** command disables only the commands specified by these arguments.

To boot the system, perform the following task in EXEC mode:

Task	Command
Boot the system.	**reload**

The following example shows how to configure the communication server to automatically boot from an image in Flash memory on a Cisco 2500:

```
cs# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
cs(config)# boot system flash igs-bfpx.102.1
cs(config)# config-register 0x2
Ctrl-Z
cs# write memory
########[ok]
cs# reload
[confirm]

System Bootstrap, Version (3.3), SOFTWARE
Copyright (c) 1986-1993 by cisco Systems
2500 processor with 1024 Kbytes of main memory

Booting igs-bfpx.102.1 from Flash address space
F3: 3911536+96836+319604 at 0x3000060

              Restricted Rights Legend

Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.

              cisco Systems, Inc.
              1525 O'Brien Drive
              Menlo Park, California 94025

3000 Software (IGS-BFPX), Version 10.2
Copyright (c) 1986-1994 by cisco Systems, Inc.
Compiled Tue 05-Jul-94 16:14

% System running from device (System flash) being initialized.
   Setting System flash access to read-only.
SNMP Research SNMP Agent Resident Module Version 12.2.0.0
Copyright 1989, 1990, 1991, 1992, 1993, 1994 SNMP Research, Inc.
cisco 2500 (68030) processor (revision A) with 1024K/1024K bytes of memory.
Processor board serial number 01244583
X.25 software, Version 2.0, NET2, BFE and GOSIP compliant.
Bridging software.
SuperLAT software (copyright 1990 by Meridian Technology Corp).
Authorized for Enterprise software set.  (0x0)
1 Ethernet/IEEE 802.3 interface.
2 Serial network interfaces.
16 terminal lines.
32K bytes of non-volatile configuration memory.
4096K bytes of processor board System flash. (Read only mode)

Press RETURN to get started!
```

Load from a Network Server

You can configure the communication server to load a system image from a network server using TFTP, rcp or mop to copy the system image file.

To do this, the configuration register boot field must be set to the correct value. See "Modify the Configuration Register Boot Field" later in this chapter. Use the **show version** command to list the current configuration register setting.

If you do not boot from a network server using MOP and you do not specify the TFTP or rcp server, by default the system image that you specify is booted from a network server using TFTP.

Note If you are using a Sun workstation as a network server and TFTP to transfer the file, set up the workstation to enable verification and generation of UDP checksums. See the Sun documentation for details.

For increased performance and reliability, boot from a system image from a network server using rcp. The rcp protocol implementation uses the Transmission Control Protocol (TCP), which ensures reliable delivery of data. If you boot the communication server from a network server using rcp, the communication server software searches for the system image on the server relative to the directory of the remote username.You cannot explicitly specify a remote username when you issue the boot command. Instead, the host name configured for the communication server is used.

You can also boot from a compressed image on a network server. One reason to use a compressed image is to ensure that there is enough memory available for storage. On communication servers that do not contain a run-from-ROM image in EPROM, when the communication server boots software from a network server, the image being booted and the running image both must fit into memory. If the running image is large, there might not be room in memory for the image being booted from the network server.

If there is not enough room in memory to boot a regular image from a network server, you can produce a compressed software image on any UNIX platform using the **compress** command. Refer to your UNIX platform's documentation for the exact usage of the **compress** command.

To specify the loading of a system image from a network server, complete the following tasks.

Task		Command
Step 1	Enter global configuration mode from the terminal.	**configure terminal**
Step 2	Specify the system image file to be booted from a network server using TFTP, rcp, or MOP server.	**boot system** [**tftp** | **rcp**] *filename* [*ip-address*] **boot system mop** *filename* [*mac-address*] [*interface*]
Step 3	Set the configuration register to enable loading of the system image from a network server.	**config-register** *value*
Step 4	Exit global configuration mode.	**Ctrl-Z**
Step 5	Write the configuration information to nonvolatile memory.	**write memory**

In the following example, the communication server is configured to use rcp to netboot from the *testme5.tester* system image file on a network server at IP address 131.108.0.1:

```
cs1# configure terminal
cs1(config)# boot system rcp testme5.tester 131.108.0.1
Ctrl-Z
cs1# write memory
```

Load from ROM

To specify the use of the ROM system image as a backup to other boot instructions in the configuration file, complete the following tasks:

Task		Command
Step 1	Enter global configuration mode from the terminal.	**configure terminal**
Step 2	Specify use of the ROM system image as a backup image.	**boot system rom**
Step 3	Set the configuration register to enable loading of the system image from ROM.	**config-register** *value*
Step 4	Exit global configuration mode.	**Ctrl-Z**
Step 5	Save the configuration information to nonvolatile memory.	**write memory**

In the following example, the communication server is configured to boot a Flash image called *image1* first. Should that image fail, the communication server will boot the configuration file *backup1* from a network server. If that method should fail, then the system will boot from ROM.

```
cs1# configure terminal
cs1(config)# boot system flash image1
cs1(config)# boot system backup1 131.108.20.4
cs1(config)# boot system rom
Ctrl-Z
cs1# write memory
```

Use a Fault-Tolerant Boot Strategy

Occasionally, network failures make netbooting impossible. To lessen the effects of network failure, consider the following boot strategy. After Flash is installed and configured, you might want to configure the communication server to boot in the following order:

1. Boot an image from Flash.
2. Boot an image from a system filename (netboot).
3. Boot from ROM image.

This boot order provides the most fault-tolerant alternative in the netbooting environment. Use the following commands in your configuration to allow you to boot first from Flash, then from a system file, and finally from ROM:

Task		Command
Step 1	Enter global configuration mode from the terminal.	**configure terminal**
Step 2	Configure the communication server to boot from Flash memory.	**boot system flash** [*filename*]
Step 3	Configure the communication server to boot from a system filename.	**boot system** *filename* [*ip-address*]
Step 4	Configure the communication server to boot from ROM.	**boot system rom**
Step 5	Set the configuration register to enable loading of the system image from a network server or Flash.	**config-register** *value*
Step 6	Exit global configuration mode.	**Ctrl-Z**
Step 7	Save the configuration information to nonvolatile memory.	**write memory**

The order of the commands needed to implement this strategy is shown in the following example:

```
cs# configure terminal
cs(config)# boot system flash gsxx
cs(config)# boot system gsxx 131.131.101.101
cs(config)# boot system rom
Ctrl-Z
cs# write memory
[ok]
cs#
```

Using this strategy, a communication server used primarily in a netbooting environment would have three alternative sources from which to boot. These alternative sources would help cushion the negative effects of a failure with the TFTP file server and of the network in general.

Specify the Configuration File the Communication Server Loads upon Restart

Configuration files can be stored on network servers. You can configure the communication server to automatically request and receive two configuration files from the network server:

- network configuration file
- host configuration file

The first file the server attempts to load is the *network configuration* file. The network configuration file contains information that is shared among several communication servers. For example, it can be used to provide mapping between IP addresses and host names.

The second file the server attempts to load is the *host configuration* file. This file contains commands that apply to one communication server in particular. Both the network and host configuration files must reside on a network server reachable using TFTP, rcp, or MOP and be readable.

You can specify an ordered list of network configuration and host configuration filenames. The communication server scans this list until it successfully loads the appropriate network or host configuration file.

Download the Network Configuration File

To configure the communication server to download a network configuration file from a server upon restart, complete the following tasks.

Task		Command
Step 1	Enter global configuration mode from the terminal.	**configure terminal**
Step 2	Enter the network configuration filename.	**boot network mop** *filename* [*mac-address*] [*interface*] **boot network** [**tftp** | **rcp**] *filename* [*ip-address*]
Step 3	Enable the communication server to automatically load the network file upon restart.	**service config**
Step 4	Exit global configuration mode.	**Ctrl-Z**
Step 5	Save the configuration information to nonvolatile memory.	**write memory**

For step 2, if you do not specify a network configuration filename, the communication server uses the default filename *network-confg*. If you do not specify a TFTP or rcp server, the communication server assumes that the you are using TFTP to transfer the file and that the server whose IP address you specify supports TFTP.

If you configure the communication server to download the network configuration file from a network server using rcp, the communication server software searches for the system image on the server relative to the directory of the remote username. The communication server host name is used as the remote username.

You can specify more than one network configuration file. The communication server tries each of them in order until it loads one successfully. This procedure can be useful for keeping files with different configuration information loaded on a network server.

Download the Host Configuration File

To configure the communication server to download a host configuration file from a server upon restart, complete the following tasks. Step 2 is optional. If you do not specify a host configuration filename, the communication server uses its own name to form a host configuration filename by converting the communication server name to all lowercase letters, removing all domain information, and appending *-confg*. If no host name information is available, the communication server uses the default host configuration filename *cs-confg*.

Task		Command
Step 1	Enter global configuration mode from the terminal.	**configure terminal**
Step 2	Optionally, enter the host configuration filename.	**boot host mop** *filename* [*mac-address*] [*interface*] **boot host** [**tftp**] *filename* [*ip-address*]
Step 3	Enable the communication server to automatically load the host file upon restart.	**service config**
Step 4	Exit global configuration mode.	**Ctrl-Z**
Step 5	Save the configuration information to nonvolatile memory.	**write memory**
Step 6	Reset the communication server with the new configuration information.	**reload**

You can specify more than one host configuration file. The communication server tries them in order until it loads one successfully. This procedure can be useful for keeping files with different configuration information loaded on a network server.

In the following example, the communication server is configured to boot from the host configuration file *hostfile1* and from the network configuration file *networkfile1*:

```
cs1# configure terminal
cs1(config)# boot host hostfile1
cs1(config)# boot network networkfile1
cs1(config)# service config
Ctrl-Z
cs1# write memory
```

If the network server fails to load a configuration file during startup, it tries again every ten minutes (default setting) until a host provides the requested files. With each failed attempt, the network server displays a message on the console terminal. If the network server is unable to load the specified file, it displays the following message:

```
Booting host-confg… [timed out]
```

Refer to the *Troubleshooting Internetworking Systems* publication for troubleshooting procedures. If there are any problems with the configuration file pointed to in nonvolatile memory, or the configuration register is set to ignore nonvolatile memory, the communication server will enter the **setup** command facility. See the *Access and Communication Servers Getting Started Guide* for details on the **setup** command.

Change the Buffer Size for Loading Configuration Files

The buffer that holds the configuration commands is generally the size of nonvolatile memory. Complex configurations might need a larger configuration file buffer size. To change the buffer size, complete the following tasks:

Task		Command
Step 1	Enter global configuration mode from the terminal.	**configure terminal**
Step 2	Change the buffer size to use for netbooting a host or network configuration file.	**boot buffersize** *bytes*
Step 3	Exit global configuration mode.	**Ctrl-Z**
Step 4	Save the configuration information to nonvolatile memory.	**write memory**

In the following example, the buffer size is set to 50000 bytes:

```
cs1# configure terminal
cs1(config)# boot buffersize 50000
Ctrl-Z
cs1# write memory
```

Compress Configuration Files

On communication servers that are equipped with nonvolatile memory, you can compress configuration files. To compress configuration files, perform the following tasks:

Task		Command
Step 1	Install the new ROMs.	Refer to the appropriate hardware installation and maintenance publication.
Step 2	Specify that the configuration file is to be compressed.	**service compress-config**
Step 3	Enter the new configuration.	Use TFTP or rcp to copy the new configuration. If you try to load a configuration that is more than three times larger than the nonvolatile memory size, the following error message is displayed: [buffer overflow - *file-size*/*buffer-size* bytes]. or **configure terminal**
Step 4	Save the new configuration.	**write memory**

Installing new ROMs is a one-time operation, and is only necessary if you do not already have Internetwork Operating System (IOS) Release 10.2 in ROM. Before you can load a configuration file that is larger than the size of nonvolatile memory, you must issue the **service compress-config** command. The **configure terminal** command works only if you have IOS Release 10.2 boot ROMs.

Manually Load a System Image

If your communication server does not find a valid system image, or if its configuration file is corrupted at startup, and the configuration register is set to enter ROM monitor mode, the system might enter ROM monitor mode. From this mode, you can manually load a system image from Flash, from a network server file, or from ROM.

You can also enter ROM monitor mode by restarting the communication server and then pressing the Break key during the first 60 seconds of startup.

Manually Boot from Flash Memory

To manually boot from Flash memory, complete the following tasks:

Task		Command
Step 1	Restart the communication server.	**reload**
Step 2	Press the Break key during the first 60 seconds while the system is starting up.	Break
Step 3	Manually boot the communication server.	**b flash** [*filename*]

In the following example, the communication server is manually booted from Flash memory. Because no *filename* is specified, the first file in Flash memory will be loaded.

```
 > b flash
Booting igs-bfpx.102.1 from Flash address space
F3: 3911536+96836+319604 at 0x3000060

                Restricted Rights Legend
```

```
Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.

             cisco Systems, Inc.
             1525 O'Brien Drive
             Menlo Park, California 94025

3000 Software (IGS-BFPX), Version 10.2
Copyright (c) 1986-1994 by cisco Systems, Inc.
Compiled Tue 05-Jul-94 16:14

% System running from device (System flash) being initialized.
   Setting System flash access to read-only.
SNMP Research SNMP Agent Resident Module Version 12.2.0.0
Copyright 1989, 1990, 1991, 1992, 1993, 1994 SNMP Research, Inc.
cisco 2500 (68030) processor (revision A) with 1024K/1024K bytes of memory.
Processor board serial number 01244583
X.25 software, Version 2.0, NET2, BFE and GOSIP compliant.
Bridging software.
SuperLAT software (copyright 1990 by Meridian Technology Corp).
Authorized for Enterprise software set.  (0x0)
1 Ethernet/IEEE 802.3 interface.
2 Serial network interfaces.
16 terminal lines.
32K bytes of non-volatile configuration memory.
4096K bytes of processor board System flash. (Read only mode)

Press RETURN to get started!
```

In the following example, the **boot flash** command is used with the image filename *igs-bfpx.102.1*. That is the file that will be loaded.

```
> b flash igs-bfpx.102.1
Booting igs-bfpx.102.1 from Flash address space
F3: 3911536+96836+319604 at 0x3000060

              Restricted Rights Legend

Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.

              cisco Systems, Inc.
              1525 O'Brien Drive
              Menlo Park, California 94025

3000 Software (IGS-BFPX), Version 10.2
Copyright (c) 1986-1994 by cisco Systems, Inc.
Compiled Tue 05-Jul-94 16:14

% System running from device (System flash) being initialized.
   Setting System flash access to read-only.
SNMP Research SNMP Agent Resident Module Version 12.2.0.0
Copyright 1989, 1990, 1991, 1992, 1993, 1994 SNMP Research, Inc.
cisco 2500 (68030) processor (revision A) with 1024K/1024K bytes of memory.
Processor board serial number 01244583
X.25 software, Version 2.0, NET2, BFE and GOSIP compliant.
Bridging software.
SuperLAT software (copyright 1990 by Meridian Technology Corp).
Authorized for Enterprise software set.  (0x0)
1 Ethernet/IEEE 802.3 interface.
2 Serial network interfaces.
16 terminal lines.
32K bytes of non-volatile configuration memory.
4096K bytes of processor board System flash. (Read only mode)

Press RETURN to get started!
```

Manually Boot from a Network File

To manually boot from a network file, complete the following tasks in EXEC mode:

Task		Command
Step 1	Restart the communication server.	**reload**
Step 2	Press the Break key during the first 60 seconds while the system is starting up.	Break
Step 3	Manually boot the communication server.	**b** *filename* [*ip-address*]

In the following example, the communication server is manually booted from the network file *network1*:

```
> b network1
```

Manually Boot from ROM

To manually boot the communication server from ROM, complete the following steps in EXEC mode:

Task		Command
Step 1	Restart the communication server.	**reload**
Step 2	Press the Break key during the first 60 seconds while the system is starting up.	Break
Step 3	Manually boot the communication server from ROM.	**b**

In the following example, the communication server is manually booted from ROM:

```
> b
```

Configure a Communication Server as a TFTP Server

As a TFTP server host, the communication server responds to TFTP Read Request messages by sending a copy of the system image contained in ROM or one of the system images contained in Flash to the requesting host. The TFTP Read Request message must use one of the filenames that are specified in the communication server's configuration.

The following algorithm is used when deciding whether to send the ROM or Flash image:

- If the specified filename is not stored in Flash memory, the ROM image is sent.
- If the specified filename exists in Flash memory, a copy of the Flash image is sent.

To specify TFTP server operation for a communication server, complete the following tasks:

Task		Command
Step 1	Enter global configuration mode from the terminal.	**configure terminal**
Step 2	Specify TFTP server operation.	**tftp-server system** *filename* [*access-list-number*]
Step 3	Exit global configuration mode.	**Ctrl-Z**
Step 4	Save the configuration information to nonvolatile memory.	**write memory**

The TFTP session can sometimes fail. To help determine why a TFTP session failed, TFTP generates an "E" character if it receives an erroneous packet, and an "O" character if it receives an out-of-sequence packet. A period (.) indicates a timeout. The transfer session may still succeed even if TFTP generates these characters, but the output is useful for diagnosing the transfer failure. For troubleshooting procedures, refer to the *Troubleshooting Internetworking Systems* publication.

In the following example, the communication server is configured to send, via TFTP, a copy of the ROM software when it receives a TFTP read request for the file version 9.0. The requesting host is checked against access list 22.

```
tftp-server system version-9.0 22
```

Configure a Communication Server to Support Incoming rcp Requests and rsh Commands

To configure the communication server to allow users on remote systems to copy files to and from the communication server and execute commands on the communication server, perform the tasks in one of the following sections:

- Configure the Communication Server to Accept rcp Requests from Remote Users
- Configure the Communication Server to Allow Remote Users to Execute Commands using rsh

Authentication of Remote rcp and rsh Users

You configure a local authentication database to control access to the communication server by remote users. A local authentication database is similar in concept and use to a UNIX *.rhosts* file. To allow remote users to execute rcp or rsh commands on the communication server, you configure entries for those users in the communication server's authentication database.

Each entry that you configure in the authentication database identifies the local user, the remote host, and the remote user. To be allowed to remotely execute commands on the communication server, the remote user must specify all three values—the local username, the remote host name, and the remote username—and therefore must be apprised of the local username. For rsh users, you can also grant a user permission to execute privileged EXEC commands remotely.

An entry that you configure in the communication server authentication database differs from an entry in a UNIX *.rhosts* file in several ways, the most salient of which is the inclusion of a local username. Because the *.rhosts* file on a UNIX system resides in the home directory of a local user account, an entry in a UNIX *.rhosts* file does not need to include the local username. The local username is determined from the user account. Because our communication servers do not inherently support the concept of accounts, you must specify the local username along with the remote host and remote username in each authentication database entry that you configure.

You can specify the communication server host name as the username. The rcp protocol requires that the remote user—that is, the client—send the local username in each rcp request to the communication server. To make the local communication server username available to remote users, you need to communicate the username to the network administrator or the remote user. To allow a remote user to execute a command on the communication server, our rcp implementation requires that the local username sent by the remote user match the local username configured in the database entry.

The communication server software uses DNS to authenticate the remote host's name and address. Because DNS can return several valid IP addresses for a host name, the communication server software checks the address of the requesting client against all of the IP addresses for the named host returned by DNS. If the address sent by the requester is considered invalid, that is, it does not match any address listed with DNS for the host name, then the communication server software will reject the remote-command execution request.

Note that if DNS support has been disabled for the local communication server, then the communication server cannot authenticate the host in this manner. In this case, IOS software sends a broadcast request attempting to gain access to DNS services on another server. If DNS services are not available, the IOS software will accept the request to remotely execute a command *only if* all three values sent with the request match exactly the values configured for an entry in the local authentication file. To bypass DNS authentication, you must give the **no ip domain-lookup** command.

To ensure security, the communication server is not enabled to support rcp requests from remote users by default. When the communication server is not enabled to support rcp, the authorization database has no effect.

Configure the Communication Server to Accept rcp Requests from Remote Users

To configure the communication server to support incoming rcp requests, complete the following tasks:

Task		Command
Step 1	Enter configuration mode from the terminal.	**configure terminal**
Step 2	Create an entry in the local authentication database for each remote user who is allowed to execute rcp commands on the communication server.	**rcmd remote-host** *local-username* {*ip-address* | *host*} *remote-username*
Step 3	Enable the communication server to support incoming rcp requests.	**rcp-enable**
Step 4	Exit configuration mode.	**Ctrl-Z**

To disable the communication server from supporting incoming rcp requests, use the **no rcp-enable** command.

Note When the communication server's support for incoming rcp requests is disabled, you can still use the rcp commands to copy images from remote servers. The communication server's support for incoming rcp requests is distinct from its capacity for outgoing rcp requests.

The following example shows how to add two entries for remote users to the communication server's authentication database and then enable the communication server to support remote copy requests from remote users. The users, named *netadmin1* on the remote host at IP address 131.108.15.555 and *netadmin3* on remote host at IP address 131.108.101.101, are both allowed to connect to the communication server and remotely execute rcp commands on it after the communication server is enabled to support rcp. Both authentication database entries give the communication server's host name *cs1* as the local username. The fourth command enables the communication server to support for rcp requests from remote users.

```
configure terminal
rcmd remote-host cs1 131.108.15.55 netadmin1
rcmd remote-host cs1 131.108.101.101 netadmin3
rcp-enable
```

Configure the Communication Server to Allow Remote Users to Execute Commands using rsh

To configure the communication server the communication server to allow remote users to execute remote shell commands using rsh on the communication server, complete the following tasks:

Task		Command
Step 1	Enter configuration mode from the terminal.	**configure terminal**
Step 2	Create an entry in the local authentication database for each remote user who is allowed to execute rsh commands on the communication server.	**rcmd remote-host** *local-username* {*ip-address* | *host*} *remote-username* [**enable**]
Step 3	Enable the communication server to support incoming rsh commands.	**rsh-enable**
Step 4	Exit configuration mode.	**Ctrl-Z**

To disable the communication server from supporting incoming rsh commands, use the **no rsh-enable** command.

Note When the communication server is disabled, you can still issue a remote shell command to be executed on other communication servers that support the remote shell protocol and on UNIX hosts on the network.

The following example shows how to add two entries for remote users to the communication server's authentication database and how to enable the communication server to support rsh commands from remote users. The users, named *rmtnetad1* and *netadmin4*, are both on the remote host at IP address 131.108.101.101. Although both users are on the same remote host, you must include a unique entry for each user. Both users are allowed to connect to the communication server and remotely execute rsh commands on it after the communication server is enabled for rsh. The user named *netadmin4* is allowed to execute privileged EXEC mode commands on the communication server. Both authentication database entries give the communication server's host name cs1 as the local username. The fourth command enables the communication server for to support rsh commands issued by remote users.

```
configure terminal
rcmd remote-host cs1 131.108.101.101 rmtnetad1
rcmd remote-host cs1 131.108.101.101 netadmin4 enable
rsh-enable
```

Configure a Communication Server as a RARP Server

You can configure the communication server as a RARP server. With this feature, RARP requests can be answered by the communication server, thereby allowing the communication server to make possible diskless booting of various systems, such as Sun workstations or PCs, on networks where the client and server are on separate subnets.

To configure the communication server as a RARP server, perform the following task in interface configuration mode:

Task	Command
Configure the communication server as a RARP server.	**ip rarp-server** *ip-address*

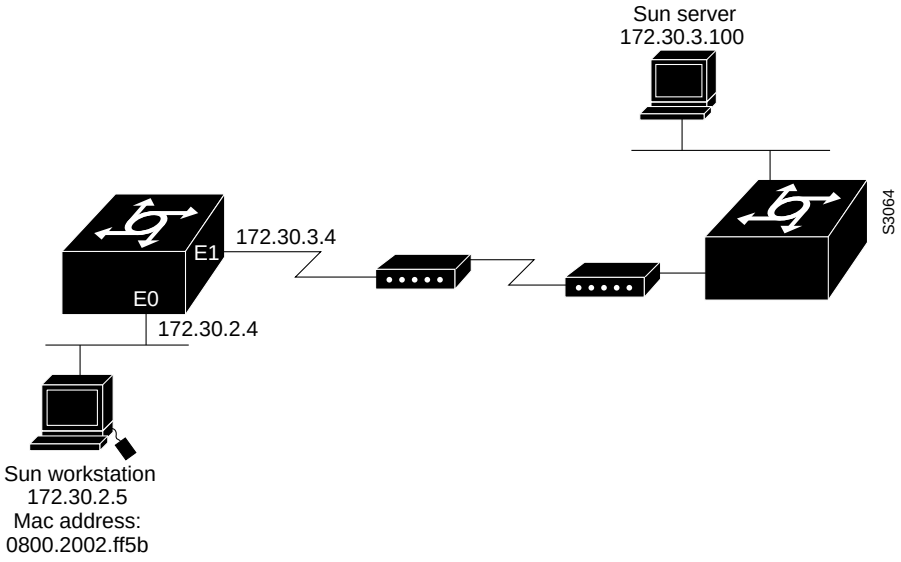

Figure 3-4 Configuring a Communication Server as a RARP Server

In the following example, the communication server is configured to act as a RARP server. Figure 3-4 illustrates the network configuration.

```
! Allow the communication server to forward broadcast portmapper requests
ip forward-protocol udp 111
! Provide the communication server with the IP address of the diskless sun
arp 128.105.2.5 0800.2002.ff5b arpa
interface ethernet 0
! Configure the communication server to act as a RARP server, using the Sun Server's
! IP address in the RARP response packet.
ip rarp-server 128.105.3.100
! Portmapper broadcasts from this interface are sent to the Sun Server.
ip helper-address 128.105.3.100
```

The Sun client and server machine's IP addresses must use the same major network number due to a limitation of the current SunOS *rpc.BOOTParamd* daemon.

Configure the Remote Username for rcp Requests

From the communication server, you can use rcp to remotely copy files to and from network servers and hosts if those systems support rcp. You do not need to configure the communication server to issue remote copy requests from the communication server using rcp. However, to prepare to use rcp from the communication server for remote copying, you can perform an optional configuration process to specify the remote username to be sent on each rcp request.

The rcp protocol requires that a client send the remote username on an rcp request. By default, the communication server software sends the remote username associated with the current TTY (terminal)process, if that name is valid, for rcp commands.

Note For UNIX systems, each physical device is represented in the file system. Terminals, or serial lines, are called TTY devices (which stands for teletype, the original UNIX terminal).

If the username for the current TTY process is not valid, the communication server software sends the host name as the remote username. For boot commands using rcp, the communication server software sends the communication server host name by default. You cannot explicitly configure the remote username.

When copying from the remote server, rcp searches for the system image or configuration file to be copied relative to the directory of the remote username. When copying to the remote server, rcp writes the system image or configuration file to be copied relative to the directory of the remote username. When booting an image, rcp searches for the image file on the remote server relative to the directory of the remote username.

To override the default remote username sent on rcp requests, complete the following tasks:

Task		Command
Step 1	Enter configuration mode from the terminal.	**configure terminal**
Step 2	Specify the remote username	**rcmd remote-username** *username*
Step 3	Exit configuration mode.	**Ctrl-Z**

To remove the remote username and return to the default value, use the **no rcmd remote-username** command.

Specify SLIP Extended BOOTP Requests

The Boot Protocol (BOOTP) server for Serial Line Internet Protocol (SLIP) supports the extended BOOTP requests specified in RFC 1084. The following command is useful in conjunction with using the auxiliary port as an asynchronous interface. To configure extended BOOTP requests for SLIP, perform the following task in global configuration mode:

Task	Command
Configure extended BOOTP requests for SLIP.	**async-bootp** *tag* [:*hostname*] *data*

You can display the extended BOOTP requests by performing the following task in EXEC mode:

Task	Command
Show parameters for BOOTP requests.	**show async bootp**

Specify MOP Server Boot Requests

To change the communication server's parameters for retransmitting boot requests to a MOP server, complete the following tasks:

Task		Command
Step 1	Enter global configuration mode from the terminal.	**configure terminal**
Step 2	Specify the code for the MOP server.	**mop device-code** {**cisco** | **ds200**}

Step 3	Set the length of time that the communication server waits before retransmitting a message.	**mop retransmit-timer** *seconds*
Step 4	Specify the number of times a communication server retransmits MOP boot requests.	**mop retries** *count*
Step 5	Exit global configuration mode.	**Ctrl-Z** or **exit**
Step 6	Save the configuration information to nonvolatile memory.	**write memory**

By default, when the communication server transmits a request that requires a response from a MOP boot server and the server does not respond, the message will be retransmitted after four seconds. If the MOP boot server and communication server are separated by a slow serial link, it might take longer than four seconds for the communication server to receive a response to its message. Therefore, you might want to configure the communication server to wait longer than four seconds before retransmitting the message if you are using such a link.

In the following example, if the MOP boot server does not respond within 10 seconds after the communication server sends a message, the communication server will retransmit the message:

```
mop retransmit-timer 10
```

Copy System Images from a Network Server to Flash Memory Using TFTP

You can copy a system image from a network server to Flash memory using TFTP by completing the following tasks:

Task		Command
Step 1	Make a backup copy of the current system software image.	See the instructions in the section "Copy System Images from Flash Memory to a Network Server Using TFTP" later in this chapter.
Step 2	Copy a system image to Flash memory.	**copy tftp flash**
Step 3	When prompted, enter the IP address or domain name of the server.	*ip-address* or *name*
Step 4	When prompted, enter the filename of the server system image.	*filename*

Note Be sure there is ample space available before copying a file to Flash. Use the **show flash** command and compare the size of the file you want to copy to the amount of available Flash memory shown. If the space available is less than the space required by the file you want to copy, the copy process will continue, but the entire file will not be copied into Flash. A failure message, "*buffer overflow - xxxx/xxxx*," will appear, where *xxxx/xxxx* is the number of bytes read in/number of bytes available.

The server system image copied to the Flash memory of the Cisco 2500 Series access server must be at least Software Version 9.21 or above.

After you issue the **copy tftp flash** command, the system prompts you for the IP address (or domain name) of the server. This can be another communication server serving ROM or Flash system software images. You are then prompted for the filename of the software image and when there is

free space available in Flash memory, you are given the option of erasing the existing Flash memory before writing onto it. If no free Flash memory space is available, or if the Flash memory has never been written to, the erase routine is required before new files can be copied. The system will inform you of these conditions and prompt you for a response. Note that the Flash memory is erased at the factory before shipment.

If you attempt to copy a file into Flash memory that is already there, a prompt will tell you that a file with the same name already exists. This file is “deleted” when you copy the new file into Flash. The first copy of the file still resides within Flash memory, but is rendered unusable in favor of the newest version, and will be listed with the [deleted] tag when you use the **show flash** command. If you abort the copy process, the newer file will be marked [deleted] because the entire file was not copied and is, therefore, not valid. In this case, the original file in Flash memory is valid and available to the system.

Following is sample output (copying a system image named *igs-bfpx.102.1*) of the prompt you will see when using the **copy tftp flash** command when Flash memory is too full to copy the file. The filename *igs-bfpx.102.1* can be in either lowercase or uppercase; the system will see *IGS-BFPX.102.1* as *igs-bfpx.102.1*. If more than one file of the same name is copied to Flash, regardless of case, the last file copied will become the valid file.

```
env-chassis# copy tftp flash
IP address or name of remote host [255.255.255.255]? dirt
Translating "DIRT"...domain server (255.255.255.255) [OK]

Name of file to copy ? igs-bfpx.102.1
Copy igs-bfpx.102.1 from 131.108.13.111 into flash memory? [confirm]
Flash is filled to capacity.
Erasure is needed before flash may be written.
Erase flash before writing? [confirm]
Erasing flash EPROMs bank 0

Zeroing bank...zzzzzzzzzzzzzzzz
Verify zeroed...vvvvvvvvvvvvvvvv
Erasing bank...eeeeeeeeeeeeeeee

Erasing flash EPROMs bank 1

Zeroing bank...zzzzzzzzzzzzzzzz
Verify zeroed...vvvvvvvvvvvvvvvv
Erasing bank...eeeeeeeeeeeeeeee

Erasing flash EPROMs bank 2

Zeroing bank...zzzzzzzzzzzzzzzz
Verify zeroed...vvvvvvvvvvvvvvvv
Erasing bank...eeeeeeeeeeeeeeee

Erasing flash EPROMs bank 3

Zeroing bank...zzzzzzzzzzzzzzzz
Verify zeroed...vvvvvvvvvvvvvvvv
Erasing bank...eeeeeeeeeeeeeeee

Loading from 131.108.1.111:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!
 [OK - 1906676/4194240 bytes]
Verifying via checksum...
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
```

```
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvv
Flash verification successful. Length = 1906676, checksum = 0x12AD
```

Note If you enter **n** after the “Erase flash before writing?” prompt, the copy process continues. If you enter **y**, the erase routine begins. Make certain you have ample Flash memory space before entering **n** at the erasure prompt.

Following is sample output from copying a system image named *igs-bfpx.102.1* into the current Flash configuration, in which a file of the name *igs-bfpx.102.1* already exists:

```
env-chassis# copy tftp flash
IP address or name of remote host [131.108.13.111]?
Name of file to copy ? igs-bfpx.102.1
File igs-bfpx.102.1 already exists; it will be invalidated!
Copy igs-bfpx.102.1 from 131.108.13.111 into flash memory? [confirm]
2287500 bytes available for writing without erasure.
Erase flash before writing? [confirm]n
Loading from 131.108.1.111:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 1906676/2287500 bytes]
Verifying via checksum...
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv
vvvvvvvvvvvvvvvvvvvvvvvvvvvvv
Flash verification successful. Length = 1902192, checksum = 0x12AD
```

In the following example, the Flash security jumper is not installed, so you cannot write files to Flash memory.

```
Everest# copy tftp flash
Flash: embedded flash security jumper(12V)
       must be strapped to modify flash memory
```

Note To abort this copy process, press **Ctrl-^** (the Ctrl, Shift, and 6 keys on a standard keyboard) simultaneously. Although the process will abort, the partial file copied before the abort was issued will remain until the entire Flash memory is erased. Refer to the *Troubleshooting Internetworking Systems* publication for procedures on how to resolve Flash memory problems.

You can copy normal or compressed images to Flash memory. You can produce a compressed system image on any UNIX platform using the **compress** command. Refer to your UNIX platform’s documentation for the exact usage of the **compress** command.

The following example shows sample output from copying a system image named *IJ09140Z* into the current Flash configuration.

```
cs# copy tftp flash
IP address or name of remote host [255.255.255.255]? server1
Name of tftp filename to copy into flash []? IJ09140Z
copy IJ09140Z from 131.131.101.101 into flash memory? [confirm] <Return>
xxxxxxxx bytes available for writing without erasure.
erase flash before writing? [confirm] <Return>
Clearing and initializing flash memory (please wait)####...
Loading from 101.2.13.110: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!... [OK - 324572/524212 bytes]
Verifying checksum...
VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV...
Flash verification successful. Length = 1204637, checksum = 0x95D9
```

The series of pound signs (#) indicates that each Flash device is being cleared and initialized; one per device. Different communication server platforms use different ways of indicating that Flash is being cleared. The exclamation points (!) indicate the copy process. The series of Vs indicates that a checksum is calculated. An O would have indicated an out-of-order packet. A period (.) would have indicated a timeout. The last line in the sample configuration indicates that the copy is successful.

Copy System Images from a Network Server to Flash Memory Using rcp

You can copy a system image from a network server to Flash memory using rcp. For the rcp command to execute properly, an account must be defined on the network server for the remote username. You can override the default remote username sent on the rcp copy request by configuring the remote username. For example, if the system image resides in the home directory of a user on the server, you can specify that user's name as the remote username. The rcp protocol implementation copies the system image from the remote server relative to the directory of the remote username.

To copy a system image from an rcp server to Flash memory, complete the following tasks:

Tasks		Command
Step 1	Make a backup copy of the current system software image.	See the instructions in the section "Copy System Images from Flash Memory to a Network Server" in this chapter.
Step 2	Enter global configuration mode from the terminal. This step is only required to override the default remote username (see step 3).	**configure terminal**
Step 3	Specify the remote username. This step is optional, but recommended.	**rcmd remote-username** *username*
Step 4	Exit global configuration mode.	**Ctrl-Z**
Step 5	Copy the system image from the network server to Flash memory using rcp.	**copy rcp flash**
Step 6	When prompted, enter the IP address or domain name of the network server.	*ip-address* or *name*
Step 7	When prompted, enter the filename of the server system image to be copied.	*filename*

Note Be sure there is ample space available before copying a file to Flash. Use the **show flash** command and compare the size of the file you want to copy to the amount of available Flash memory shown. If the space available is less than the space required by the file you want to copy, the copy process will continue, but the entire file will not be copied into Flash. A failure message, "*buffer overflow - xxxx/xxxx*," will appear, where *xxxx/xxxx* is the number of bytes read in/number of bytes available.

The server system image copied to the Flash memory of the ASM-CS must be at least Software Version 9.0 or above. For the Cisco 2500 series, the server system image must be at least Software Version 9.21 or above.

When you issue the **copy rcp flash** command, the system prompts you for the IP address (or domain name) of the server. This can be another communication server serving ROM or Flash system software images. You are then prompted for the filename of the software image; when there is free space available in Flash memory, you are given the option of erasing the existing Flash memory before writing onto it. If no free Flash memory space is available, or if the Flash memory has never been written to, the erase routine is required before new files can be copied. The system will inform you of these conditions and prompt you for a response. If you accept the erasure, the system will prompt you again to confirm before erasing. Note that the Flash memory is erased at the factory before shipment.

If you attempt to copy a file into Flash memory that is already there, a prompt will tell you that a file with the same name already exists. This file is "deleted" when you copy the new file into Flash. The first copy of the file still resides within Flash memory, but is rendered unusable in favor of the newest version, and will be listed with the [deleted] tag when you use the **show flash** command. If you abort the copy process, the newer file will be marked [deleted] because the entire file was not copied and is, therefore, not valid. In this case, the original file in Flash memory is valid and available to the system.

The following example copies a system image named *IJ09140z* from the *netadmin1* directory on the remote server named *SERVER1.CISCO.COM* with an IP address of 131.131.101.101 to the communication server's Flash memory. To ensure that enough Flash memory is available to accommodate the system image to be copied, the communication server software allows you to erase the contents of Flash memory first.

```
cs1# configure terminal
cs1# rcmd remote-username netadmin1
Ctrl-Z
cs1# copy rcp flash
System flash directory:
File name/status
 1 IJ09140Z
[2076072 bytes used, 21080 bytes available]

Address or name of remote host[UNKNOWN]? 131.131.101.101
Name of file to copy? IJ09140Z
Copy IJ09140z from SERVER1.CISCO.COM?[confirm]

Checking for file 'IJ09140Z' on SERVER1.CISCO.COM...[OK]

Erase flash device before writing?[confirm]
Are you sure?[confirm]
Erasing device...ezeeze...erased.

Connected to 131.131.101.101

Loading 2076007 byte file IJ09140Z:
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!![OK]

Verifying checksum... (0x87FD)...[OK]
cs1#
```

The exclamation points (!) indicate that the copy process is taking place. Each exclamation point (!) indicates that ten packets have been transferred successfully.

Note If you enter **n** after the "Erase flash device before writing?" prompt, the copy process continues. If you enter **y** and you confirm the erasure, the erase routine begins. Make certain you have ample Flash memory space before entering **n** at the erasure prompt.

You can copy normal or compressed images to Flash memory. You can produce a compressed system image on any UNIX platform using the **compress** command. Refer to your UNIX platform's documentation for the exact usage of the **compress** command.

Use Flash Load Helper

Flash load helper is a software option available to users who want to upgrade their system software on run-from-Flash systems. Flash load helper simplifies the upgrade procedure without requiring additional hardware; however, it does require some brief network downtime. A system image running from Flash can use Flash load helper only if the boot ROMs support Flash load helper. If the boot ROMs do not support Flash load helper, you must perform the Flash upgrade manually.

Flash load helper is an automated procedure that reloads from the current running image to the ROM-based bootstrap image, downloads to Flash memory, and reboots to the system image in Flash memory. Flash load helper performs checks and validations to maximize the success of a Flash upgrade and minimize the chance of leaving Flash memory in an erased state or with a file that cannot boot.

In run-from-Flash systems, the software image in the communication server is stored in and executed from the Flash EPROM, rather than being executed from RAM, thereby reducing memory cost. A run-from-Flash system requires enough Flash EPROM to hold the image and enough main system RAM to hold the routing tables and data structures. The system does not need the same amount of main system RAM as a run-from-RAM system because the full image does not reside in RAM. Run-from-flash systems include the Cisco 2500 series.

Flash load helper includes the following features:

- Confirms access to the specified source file on the specified server before erasing Flash memory and reloading to the ROM image for the actual upgrade.
- Warns you if the image being downloaded is not appropriate for the system.
- Prevents reloads to the ROM image for a Flash upgrade if the system is not set up for autobooting and the user is not on the console terminal. Then if a catastrophic failure occurs during the upgrade, you can bring up the boot ROM image as a last resort rather than have the system wait at the ROM monitor's prompt for input from the console terminal.
- Retries Flash downloads automatically up to six times. The retry sequence is as follows:
  - First try

  - Immediate retry
  - Retry after 30 seconds
  - Reload ROM image and retry
  - Immediate retry
  - Retry after 30 seconds
- Allows you to save any configuration changes made before they exit out of the system image.
- Notifies users logged into the system of the impending switch to the boot ROM image so that they do not lose their connections unexpectedly.
- Logs console output during the Flash load helper operation into a buffer that is preserved through system reloads. You can retrieve the buffer contents from a running image. The output can be useful if console access is unavailable or a failure occurs in the download operation.

To download a new file to Flash memory, perform the following tasks in EXEC mode:

Task	Command
Download a file to Flash memory.	**copy tftp flash**

Flash load helper also supports the **copy mop flash** command. However, the **copy mop flash** command does not provide all the enhanced features available in the **copy tftp flash** command. Specifically, it does not provide the access check for the file on the MOP server, the size check to ensure that the file will fit into Flash memory, or warnings if the file is not appropriate for the system. Another difference between the **copy tftp flash** command and the **copy mop flash** command is that there is no prompt for the MOP server address (similar to the TFTP server address prompt), because the MOP server is automatically solicited. Other enhanced features of the **copy mop flash** command are identical to the **copy tftp flash** command enhanced features.

Upgrade System Software Using Flash Load Helper

This section describes how to upgrade system software using Flash load helper. To download a new file to Flash memory, perform the following tasks in EXEC mode:

Task	Command
Download a file to Flash memory.	**copy tftp flash**

As long as the boot ROMs support Flash load helper, executing the **copy tftp flash** command automatically invokes Flash load helper.

You can always invoke Flash load helper from a console terminal. You can also invoke Flash load helper from a virtual terminal (for example, a Telnet session) if the system is configured for autobooting. This means that the boot bits in the system configuration register must be nonzero. Refer to the appropriate hardware installation manual for information about setting the boot bits.

```
cs# copy tftp flash
ERR: Config register boot bits set for manual booting
```

The error message “ERR: Config register boot bits set for manual booting” displays if you are in a Telnet session and the system is set for manual booting (the boot bits in the configuration register are zero). If you were to invoke Flash load helper when the system is set for manual booting, the system might enter ROM monitor mode in case of any catastrophic failure in the Flash upgrade, thus taking it out of the remote Telnet user’s control. The system would try to bring up at least the boot

ROM image if it could not boot an image from Flash memory. Use the **config-register** global configuration command to change the configuration register value so that the boot bits are nonzero before reinitiating the **copy tftp flash** command.

The **copy tftp flash** command initiates a dialog similar to the following:

```
cs# copy tftp flash

************************** NOTICE ******************************
Flash load helper v1.0
This process will accept the TFTP copy options and then terminate the current system image
to use the ROM based image for the copy. Router functionality will not be available during
that time. If you are logged in via telnet, this connection will terminate. Users with
console access can see the results of the copy operation.
****************************************************************
```

If any terminals other than the one on which this command is being executed are active, the following message appears:

```
There are active users logged into the system.

Proceed? [confirm] y
System flash directory:
File Length  Name/status
1    2251320 abc/igs-kf.914
[2251384 bytes used, 1942920 available, 4194304 total]
```

Enter the IP address or name of the remote host you are copying from:

```
Address or name of remote host [255.255.255.255]? 131.108.1.111
```

Enter the name of the file you want to copy:

```
Source file name? abc/igs-kf.914
```

Enter the name of the destination file:

```
Destination file name [default = source name]? <Return>
Accessing file 'abc/igs-kf.914' on 131.108.1.111....
Loading from 131.108.13.111:
Erase flash device before writing? [confirm] <Return>
```

If you choose to erase Flash memory, the dialog continues as follows. The **copy tftp flash** operation verifies the request from the running image by trying to TFTP a single block from the remote TFTP server. Then Flash load helper is executed, causing the system to reload to the ROM-based system image.

```
Erase flash device before writing? [confirm] y
Flash contains files. Are you sure? [confirm] y
```

If the file does not seem to be a valid image for the system, a warning appears; you must issue confirmation.

```
Copy 'abc/igs-kf.914' from TFTP server
as 'abc/igs-kf.914' into Flash WITH erase? y

%SYS-5-RELOAD: Reload requested
%FLH: rxboot/igs-kf.914r from 131.108.1.111 to flash ...
```

If you choose not to erase Flash memory and there is no file duplication, the dialog would have continued as follows:

```
Erase flash device before writing? [confirm] n
Copy 'abc/igs-kf.914' from TFTP server
as 'abc/igs-kf.914' into Flash WITHOUT erase? y
```

If you choose not to erase Flash memory, and there was file duplication, the dialog would have continued as follows:

```
Erase flash device before writing? [confirm] n
File 'abc/igs-kf.914' already exists; it will be invalidated!
Invalidate existing copy of 'abc/igs-kf' in flash memory? [confirm] y
Copy 'abc/igs-kf.914' from TFTP server
as 'abc/igs-kf.914' into Flash WITHOUT erase? y
```

If the configuration has been modified but not yet saved, you are prompted to save the configuration:

```
System configuration has been modified. Save? [confirm]
```

If you confirm to save the configuration, you might also receive this message:

```
Warning: Attempting to overwrite an NVRAM configuration previously written by a different
version of the system image. Overwrite the previous NVRAM configuration? [confirm]
```

Users with open Telnet connections are notified of the system reload, as follows:

```
**System going down for Flash upgrade**
```

In case of TFTP failures, the copy operation is retried up to three times. If the failure happens in the middle of a copy (part of the file has been written to Flash memory), the retry does not erase Flash memory unless you specified an erase. The partly written file is marked as deleted and a new file is opened with the same name. If Flash memory runs out of free space in this process, the copy is terminated.

After Flash load helper finishes its copy (whether successful or not), it automatically attempts an autoboot or a manual boot, depending on the value of the boot bits in the configuration register. If the boot bits are zero, the system attempts a default boot from Flash memory (equivalent to a manual **b flash** command at the ROM monitor prompt) to load up the first bootable file in Flash memory.

If the boot bits are nonzero, the system attempts to boot based on the boot configuration commands. If no boot configuration commands exist, the system attempts to load the first bootable file in Flash memory.

Monitor Flash Load Helper

To view the system console output generated during the Flash load helper operation, perform the following task in EXEC mode:

Task	Command
View the system console output generated by Flash load helper.	**show flh-log**

Verify the Image in Flash Memory

When you issue the **copy tftp flash**, **copy rcp flash**, or **copy rcp bootflash** commands, the checksum of the image in Flash memory is displayed at the bottom of the screen. Before booting from Flash memory, verify that the checksum of the image in Flash memory matches the checksum listed in the README file that was distributed with the system software image. The README file was copied to the network server automatically when you installed the system software image on the server.

Caution If the checksum value does not match the value in the README file, do not reboot the communication server. Issue the copy request and compare the checksums again. If the checksum is repeatedly wrong, copy the original system software image bootstrap image back into Flash memory *before* you reboot the communication server from Flash memory. If you have a bad image in Flash memory and try to boot from Flash, the communication server will start the system image contained in ROM (assuming netbooting is not configured). If ROM does not contain a fully functional system image, the communication server will not function and will have to be reconfigured through a direct console port connection.

Use Dual Flash Bank

Dual Flash bank is a software feature that allows you to partition the two banks of Flash memory into two separate, logical devices so that each logical device has its own file system. This feature is available on Cisco 2500 series systems with at least two banks of Flash memory; one bank is one SIMM. The minimum partition size is the size of a bank.

Because they must be capable of accessing files in any file system, boot ROMs must be one of the following versions:

- 9.14(8) or higher
- 10.0(5) or higher

There are several benefits to partitioning Flash memory:

- For all systems, partitioning provides a better way to manage files in Flash memory, especially if the Flash memory size is large.
- For systems that execute code out of Flash memory, partitioning allows you to download a new image into the file system in one Flash memory bank while an image is being executed from the file system in the other bank. The download is simple and causes no network disruption or downtime. After the download is complete, you can switch to the other bank at a convenient time.
- One system can hold two different images, with one image acting as a backup for the other. Then, if a downloaded image fails to boot for some reason, the earlier running, good image is still available. Each bank is treated as a separate device.

You might use Flash load helper rather than dual Flash bank for one of the following reasons:

- If you want to download a new file into the bank from which the current system image is executing
- If you want to download a file that is larger than the size of a bank, and hence want to switch to only a single bank mode

To use dual Flash bank, perform the tasks in one or more of the following sections:

- Understand Relocatable Images
- Upgrade to Release 10.2
- Partition Flash Memory
- Download a File into a Flash Partition
- Manually Boot from Flash Memory
- Configure the Communication Server to Boot Automatically from Flash Memory
- Configure a Flash Partition as a TFTP Server

Understand Relocatable Images

Partitioning requires that run-from-Flash images be loaded into different Flash memory banks at different physical addresses. This means that images must be relocatable. A relocatable image is an image that contains special relocation information that allows the following:

- The image to relocate itself whenever it is loaded into RAM for execution
- A download program with appropriate support to relocate the image before storage in Flash memory, so that the image can run in place in Flash memory, regardless of where in Flash memory it is stored

Run-from-Flash systems running norelocatable images execute images that need to be stored in Flash memory at a specific address. This means storing the image as the first file in Flash. If the image is stored at any other location in Flash, it cannot be executed, nor can the image be executed from RAM. The relocatable image is necessary to overcome this limitation.

With Flash partitioning, the nonrelocatable run-from-Flash images will not work unless loaded into the first device as the first file. This requirement defeats the purpose of partitioning. However, relocatable images can be loaded into any Flash partition (and not necessarily as the first file within the partition) and executed in place.

Note that unless downloaded as the first file in the first partition, this download must be done by an image that recognizes relocatable images.

A relocatable image is an image created for Release 10.0(6), 10.2(2), or later. A nonrelocatable image is an image that was created before this software release and hence does not recognize relocatable images. The following are nonrelocatable images:

- Any image from a release prior to Release 10.0
- Any 10.0 image prior to Release 10.0(6)
- Release 10.2(1)

You can identify relocatable image by its name. The naming convention for image names for storage on a UNIX system is as follows:

platform-capabilities-type

The letter “l” in the *type* field indicates a relocatable image. The following are examples of some relocatable image names:

- igs-i-l—IP image
- igs-d-l—Desktop image
- igs-bpx-l—Enterprise image

Only the “igs” prefix images are available as relocatable images. Images distributed on floppy diskettes might have different naming conventions.

For backward compatibility, the relocatable images have been linked to execute as the first file in the first Flash memory bank. This makes the images similar to previous Flash images. Thus, if you download a relocatable image into a nonrelocatable image system, the image will run correctly from Flash memory.

Upgrade to Release 10.2

If you upgrade to IOS Release 10.2 from a previous software release, you need to erase Flash when you are prompted during the download. This is to ensure that the image is downloaded as the first file in Flash memory.

Partition Flash Memory

To partition Flash memory, perform the following task in global configuration mode:

Task	Command
Partition Flash memory.	**partition flash** *partitions* [*size1 size2*]

This task will succeed only if the system has at least two banks of Flash memory and the partitioning does not cause an existing file in Flash memory to be split across the two partitions.

Download a File into a Flash Partition

To download a file into a Flash partition, perform one of the following tasks in EXEC mode:

Task	Command
Download a file from a TFTP server into a Flash partition.	**copy tftp flash**
Download a file from a MOP server into a Flash partition.	**copy mop flash**
Download a file from an rcp server into a Flash partition.	**copy rcp flash**

The prompts displayed after you execute the **copy tftp flash**, **copy mop flash**, or **copy rcp flash** command indicate the method by which the download can be done into each partition. The possible methods are as follows:

- None—There is no way to copy into the partition.
- RXBOOT-Manual—You must manually reload to the rxboot image in ROM in order to copy the image.
- RXBOOT-FLH—The copy will be done using the Flash load helper software in boot ROM; that is, it will be done automatically.
- Direct—The copy can be done directly.

If the image can be downloaded into more than one partition, you are prompted for the partition number. Enter one of the following at the partition number prompt to obtain help:

- **?**—Display the directory listings of all partitions.
- **?1**—Display the directory of the first partition.
- **?2**—Display the directory of the second partition.
- **q**—Quit the copy command.

Manually Boot from Flash Memory

To manually boot the communication server from Flash memory, perform one of the following tasks in ROM monitor mode:

Task	Command
Boot the first bootable file found in any partition.	**b flash** or **b flash flash:**

Task	Command
Boot the first bootable file from the specified partition.	**b flash** *partition-number***:** or **b flash flash:***partition-number***:**
Boot a file from the first partition.	**b flash** *filename* or **b flash flash:***filename*
Boot a file from the specified partition.	**b flash** *partition-number***:***filename* or **b flash flash:***partition-number***:***filename*

The result of booting a relocatable image from Flash memory depends on where and how the image was downloaded into Flash memory. For a definition of relocatable and nonrelocatable images, see the section "Understand Relocatable Images" in this chapter. Table 3-2 describes various means by which an image could be downloaded and the corresponding result of booting from Flash memory.

Table 3-2 Downloading an Image and Booting from Flash Memory

Download Method	Result of Booting from Flash Memory
The image was downloaded as the first file by a nonrelocatable image.	The image will execute in place from Flash memory, just like a run-from-Flash image.
The image was downloaded as a subsequent file by a nonrelocatable image.	The nonrelocatable image would not have relocated the image before storage in Flash memory. This image will not be booted.
The image was downloaded as the first file by a relocatable image.	The image will execute in place from Flash memory, just like a run-from-Flash image.
The image was downloaded as a subsequent file by a relocatable image (possibly into the second partition).	The relocatable image relocates the image before storage in Flash memory. Hence, the image will execute in place from Flash memory, just like any other run-from-Flash image.

Configure the Communication Server to Boot Automatically from Flash Memory

To configure the communication server to boot automatically from Flash memory, perform one of the following tasks in global configuration mode:

Task	Command
Boot the first bootable file found in any partition.	**boot system flash** or **boot system flash:**
Boot the first bootable file from the specified partition.	**boot system flash** *partition-number***:** or **boot system flash flash:***partition-number***:**
Boot a file from the first partition.	**boot system flash** *filename* or **boot system flash flash:***filename*
Boot a file from the specified partition.	**boot system flash** *partition-number***:***filename* or **boot system flash flash:***partition-number***:***filename*

The result of booting a relocatable image from Flash memory depends on where and how the image was downloaded into Flash memory. Table 3-2, shown earlier, describes various means by which an image could be downloaded and the corresponding result of booting from Flash memory.

Configure a Flash Partition as a TFTP Server

To configure a Flash partition as a TFTP server, perform one of the following tasks in global configuration mode:

Task	Command
Specify TFTP server operation:	
file from first partition	**tftp-server system** *filename*
file from first partition	**tftp-server system flash:***filename*
file from partition number	**tftp-server system** *partition-number*:*filename*
file from partition number	**tftp-server system flash:***partition-number*:*filename*

Once you have specified TFTP server operation, exit configuration mode and save the configuration information to nonvolatile memory.

Copy System Images from Flash Memory to a Network Server Using TFTP

You can use TFTP to copy a system image back to a network server. This copy of the system image can serve as a backup copy and also can be used to verify that the copy in Flash is the same as on the original file on disk. To copy the system image to a network server, perform the following task:

Task		Command
Step 1	Learn the exact spelling of the system image in Flash memory.	**show flash** [**all**]
Step 2	Copy the system image in Flash memory to a TFTP server.	**copy flash tftp**
Step 3	When prompted, enter the IP address or domain name of the TFTP server.	*ip-address* or *name*
Step 4	When prompted, enter the filename of the system image in Flash memory.	*filename*

The following example uses the **show flash all** command to learn the name of the system image file and the **copy flash tftp** command to copy the system image to a TFTP server. The name of the system image file (*xk09140z*) is listed near the end of the **show flash all** output.

```
cs# show flash all
System flash directory:
File  name/status
        addr     length    fcksum  ccksum
  1   igs-bfpx.102.1
        0x40     4008404   0x35B3  0x35B3
[4008468 bytes used, 185836 bytes available]
4096K bytes of processor board System flash. (Read only mode)
System flash chips could not be identified.
Check the Vpp (12V) jumper installation (if present)
and/or the chips/SIMMs installed.

Flash chips supported by system :
   Code   Chip-Sz     Cmd-grp   Chip-name
   89B4   0x20000        1      INTEL 28F010
   89BD   0x40000        1      INTEL 28F020
   01A7   0x20000        1      AMD   28F010
   012A   0x40000        1      AMD   28F020
   1CD0   0x40000        1      M5M   28F101P
   89A2   0x100000       2      INTEL 28F008SA

2048K bytes of flash memory on embedded flash (in XX).
   ROM   socket    code    bytes       name
    0      U42     89BD   0x40000     INTEL 28F020
    1      U44     89BD   0x40000     INTEL 28F020
    2      U46     89BD   0x40000     INTEL 28F020
    3      U48     89BD   0x40000     INTEL 28F020
    4      U41     89BD   0x40000     INTEL 28F020
    5      U43     89BD   0x40000     INTEL 28F020
    6      U45     89BD   0x40000     INTEL 28F020
    7      U47     89BD   0x40000     INTEL 28F020
  security jumper(12V) is installed,
  flash memory is programmable.
file  offset      length       name
 0     0x40        1204637     xk09140z
  [903848/2097152 bytes free]

cs# copy flash tftp
IP address of remote host [255.255.255.255]? 101.2.13.110
filename to write on tftp host? igs-bfpx.102.1
writing igs-bfpx.102.1 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
successful tftp write.
```

To stop the copy process, press **Ctrl-**^. Refer to the *Troubleshooting Internetworking Systems* publication for procedures on how to resolve Flash memory problems.

Once you have configured Flash memory, you might want to configure the system (using the **configure terminal** command) with the **no boot system flash** configuration command to revert to booting from ROM (for example, if you do not yet need this functionality, if you choose to netboot, or if you do not have the proper image in Flash memory). After you enter the **no boot system flash** command, use the **write memory** command to save the new configuration command to nonvolatile memory.

Copy System Images from Flash Memory to a Network Server Using rcp

You can copy a system image back to a network server. This copy of the system image can serve as a backup copy and also can be used to verify that the copy in Flash is the same as on the original file on disk.

The rcp protocol requires that a client send the remote username on each rcp request to the server. When you copy a bootstrap image from Flash memory to a network server using rcp, the communication server software sends the remote username associated with the current TTY (terminal) process, if that name is valid. If the TTY remote username is invalid, the communication server software uses the communication server host name as the both the remote and local usernames.

Note For UNIX systems, each physical device is represented in the file system. Terminals, or serial lines, are called TTY devices (which stands for teletype, the original UNIX terminal).

You can configure a different remote username to be sent to the server. The rcp protocol implementation writes the system image relative to the directory associated with the remote username on the network server.

For the rcp copy command to execute properly, an account must be defined on the destination server for the remote username.

To stop the copy process, press **Ctrl-^**. Refer to the *Troubleshooting Internetworking Systems* publication for procedures on how to resolve Flash memory problems.

If you copy the system image to a personal computer used as a file server, the computer must support the remote shell protocol.

To copy the system image to a network server, perform the following tasks:

Task		Command
Step 1	Enter configuration mode from the terminal. This step is only required if you are going to override the default remote username (see step 2).	**configure terminal**
Step 2	Specify the remote username. This step is optional, but recommended.	**rcmd remote-username** *username*
Step 3	Exit configuration mode.	**Ctrl-Z**
Step 4	Using rcp, copy the system image in Flash memory to a network server.	**copy flash rcp**
Step 5	When prompted, enter the IP address or domain name of the server.	*ip-address* or *name*
Step 6	When prompted, enter the filename of the system image in Flash memory.	*filename*

The following example copies the system image *gsxx* to a network server using rcp:

```
cs# configure terminal
cs# rcmd remote-username netadmin1
Ctrl-Z
cs# copy flash rcp
System flash directory:
File name/status
 1 gsxx
[2076072 bytes used, 21080 bytes available]

Name of file to copy? gsxx
Address or name of remote host [UNKNOWN]? 131.108.1.111
File name to write to? gsxx
Verifying checksum for 'gsxx' (file # 1)...[OK]

Writing gsxx !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
cs#
```

The exclamation points (!) indicate that the copy process is taking place. Each exclamation point (!) indicates that ten packets have been transferred successfully.

Copy a Configuration File from a Network Server to the Communication Server Using rcp

You can copy a configuration file from a network server to the local communication server using rcp. You might use this process to restore a configuration file to the communication server if you have backed up the file to a server. If you replace a communication server and want to use the configuration file that you created for the original communication server, you could restore that file instead of recreating it. You can also use this process to copy to the communication server a different configuration that is stored on a network server.

There are two ways to copy a configuration file from a network server using rcp:

- Copy the file to nonvolatile memory. You can copy a configuration file from a network server to the communication server's nonvolatile memory.
- Copy and run the file. You can copy a configuration file from a network server to the communication server and run that configuration from RAM.

The rcp protocol requires that a client send the remote username on each rcp request to a network server. When you issue a request to copy a configuration file from a network server using rcp and copy it to nonvolatile memory or copy and run it, the communication server sends a default remote username unless you override the default by configuring a remote username. As the default value of the remote username, the communication server software sends the remote username associated with the current TTY process, if that name is valid. If the TTY remote username is invalid, the communication server software uses the communication server host name as the both the remote and local usernames. The rcp protocol implementation searches for the configuration file to be copied relative to the directory associated with the remote username on the network server.

For the rcp copy request to execute successfully, an account must be defined on the network server for the remote username. If you copy the configuration file from a personal computer used as a file server, the remote host computer must support the remote shell protocol.

Copy a Configuration File to Nonvolatile Memory

You can retrieve the commands stored in a configuration file on a server and write them to a file of the same name stored in nonvolatile memory on the communication server.

A host configuration file contains commands that apply to one network server in particular. A network configuration file contains commands that apply to all network servers on a network.

To copy a configuration file from a network server using rcp, perform the following tasks:

Task		Command
Step 1	Enter configuration mode from the terminal. This step is only required if you are going to override the default remote username (see step 2).	**configure terminal**
Step 2	Specify the remote username. This step is optional, but recommended.	**rcmd remote-username** *username*
Step 3	Exit configuration mode.	**Ctrl-Z**
Step 4	Using rcp, copy the configuration file from a network server to the communication server's nonvolatile memory.	**copy rcp startup-config**
Step 5	When prompted, enter the IP address of the network server.	*ip-address*
Step 6	When prompted, enter the name of the configuration file.	*filename*

The following example specifies a remote username of *netadmin1*. Then it copies a host configuration file *host2-confg* from the *netadmin1* directory on the remote server with an IP address of 131.131.101.101, and stores that file in nonvolatile memory on the communication server:

```
cs2# configure terminal
cs2# rcmd remote-username netadmin1
Ctrl-Z
cs2# copy rcp startup-config
Host or network configuration file [host]?
Address of remote host [255.255.255.255]? 131.131.101.101
Name of configuration file[rtr2-confg]?host2-confg
Configure using rtr2-confg from 131.131.101.101?[confirm]
Connected to 131.131.101.101
Loading 1112 byte file rtr2-confg:![OK]
[OK]
cs2#
%SYS-5-CONFIG_NV:Non-volatile store configured from rtr2-config by rcp from
131.131.101.101
```

Copy and Run the Configuration File

You can copy a configuration file from a network server and load and run the configuration file on the communication server.

A host configuration file contains commands that apply to one network server in particular. A network configuration file contains commands that apply to all network servers on a network.

To copy a configuration file from a network server using rcp, load the configuration file into RAM on the communication server, and run the file, perform the following tasks:

Task		Command
Step 1	Enter configuration mode from the terminal. This step is only required if you are going to override the default remote username (see step 2).	**configure terminal**
Step 2	Specify the remote username. This step is optional, but recommended.	**rcmd remote-username** *username*
Step 3	Exit configuration mode.	**Ctrl-Z**
Step 4	Using rcp, copy the configuration file from a network server to the communication server.	**copy rcp running-config**
Step 5	When prompted, enter the IP address of the server.	*ip-address*
Step 6	When prompted, enter the name of the configuration file.	*filename*

The following example copies a host configuration file named *host1-confg* from the *netadmin1* directory on the remote server with an IP address of 131.108.101.101, and loads and runs that file on the communication server:

```
cs# configure terminal
cs# rcmd remote-username netadmin1
Ctrl-Z
cs# copy rcp running-config
Host or network configuration file [host]?
Address of remote host [255.255.255.255]?131.131.101.101
Name of configuration file [cs-confg]?host1-confg
Configure using host1-confg from 131.131.101.101? [confirm]
Connected to 131.131.101.101
Loading 1112 byte file host1-confg:![OK]
cs#
%SYS-5-CONFIG: Configured from host1-config by rcp from 131.131.101.101
```

Copy a Configuration File from the Communication Server to a Network Server

You can use TFTP to copy a configuration file from the communication server to a network server. The configuration file that you copy to must already exist on the server and be globally writable before the network server allows you to write to it.

To store configuration information on a network server using TFTP, complete the following tasks in the EXEC mode:

Task		Command
Step 1	Specify that the communication server configuration file in nonvolatile memory should be stored on a network server.	**write network**
Step 2	Enter the IP address of the network server.	*ip-address*
Step 3	Enter the name of the configuration file to store on the server.	*filename*
Step 4	Confirm the entry.	**y**

The command prompts you for the destination host's address and a filename, as the following example illustrates.

The following example copies a configuration file from a communication server to a server:

```
Tokyo# write network
Remote host [131.108.2.155]?
Name of configuration file to write [tokyo-confg]?
Write file tokyo-confg on host 131.108.2.155? [confirm] y
#
Writing tokyo-confg !! [OK]
```

Copy a Configuration File from the Communication Server to a Network Server Using rcp

You can use rcp to copy configuration files from the local communication server to a network server. You can back up current configuration files before you change a file's contents, and restore the original configuration files from the server at a later time.

You can copy a startup configuration file or a running configuration file to the server.

The rcp protocol requires that a client send the remote username on each rcp request to a server. When you issue a command to copy a configuration file from the communication server to a server using rcp, the communication server sends a default remote username unless you override the default by configuring a remote username. As the default value of the remote username, the communication server software sends the remote username associated with the current TTY (terminal) process, if that name is valid.

Note For UNIX systems, each physical device is represented in the file system. Terminals, or serial lines, are called TTY devices (which stands for teletype, the original UNIX terminal).

If the TTY remote username is invalid, the communication server software uses the communication server host name as the both the remote and local usernames. The rcp protocol implementation writes the configuration file to be copied relative to the directory associated with the remote username on the server.

For the rcp copy request to execute successfully, an account must be defined on the network server for the remote username.

If you copy the configuration file to a PC used as a file server, the PC must support rsh.

Copy a Startup Configuration File to a Network Server Using rcp

You can copy the contents of the configuration file in nonvolatile memory to a network server using rcp. The copied file can serve as a backup configuration file.

To copy a startup configuration file to a network server using rcp, complete the following tasks:

Task		Command
Step 1	Enter configuration mode from the terminal. This step is only required if you are going to override the default remote username (see step 2).	**configure terminal**
Step 2	Specify the remote username. This step is optional, but recommended.	**rcmd remote-username** *username*
Step 3	Specify that the Communication server configuration file in nonvolatile memory should be copied to a network server using rcp.	**copy startup-config rcp**
Step 4	Enter the IP address of the network server.	*ip-address*
Step 5	Enter the name of the configuration file to store on the server.	*filename*
Step 6	Confirm the entry.	**y**

The following example shows how to store a startup configuration file on a server using rcp to copy the file:

```
cs# configure terminal
cs# rcmd remote-username netadmin2
Ctrl-Z
cs# copy startup-config rcp
Remote host[]? 131.131.101.101
Name of configuration file to write [rtr2-confg]?
Write file rtr2-confg on host 131.131.101.101?[confirm]
![OK]
```

Copying a Running Configuration File to a Network Server Using rcp

You can copy the running configuration file to a server using rcp. The copied file can serve as a backup configuration file.

To store a running configuration file on a server, complete the following tasks:

Task		Command
Step 1	Enter configuration mode from the terminal. This step is only required if you are going to override the default remote username (see step 2).	**configure terminal**
Step 2	Specify the remote username. This step is optional, but recommended.	**rcmd remote-username** *username*
Step 3	Specify that the communication server's running configuration file should be stored on a network server.	**copy running-config rcp**
Step 4	Enter the IP address of the network server.	*ip-address*
Step 5	Enter the name of the configuration file to store on the server.	*filename*
Step 6	Confirm the entry.	**y**

The following example copies the running configuration file named *cs2-confg* to the *netadmin1* directory on the remote host with an IP address of 131.108.101.101:

```
cs# configure terminal
cs# rcmd remote-username netadmin1
Ctrl-Z
cs# copy running-config rcp
Remote host[]? 131.131.101.101
Name of configuration file to write [cs2-confg]?
Write file rtr2-confg on host 131.131.101.101?[confirm]
###![OK]
Connected to 131.131.101.101
```

Display System Image and Configuration Information

Perform the following tasks in EXEC mode to display information about system software, system image files, and configuration files:

Task	Command
List the system software release version, configuration register setting, and so on.	**show version**
List the configuration information stored in nonvolatile memory.	**show configuration**
List the configuration information in running memory.	**write terminal**
List information about Flash memory, including system image filenames and amounts of memory used and remaining.	**show flash**
List information about Flash memory, including system image filenames, amounts of memory used and remaining, and Flash partitions.	**show flash** [**all** | **chips** | **detailed** | **err** | **partition** *number* [**all** | **chips** | **detailed** | **err**] | **summary**]
Display the parameters that have been configured for SLIP extended BOOTP requests.	**show async bootp**

You can also use the **o** command in ROM monitor mode to list the configuration register settings on some models.

The Flash content listing does not include the checksum of individual files. To recompute and verify the image checksum after the image is copied into Flash memory, complete the following task in EXEC mode:

Task	Command
Recompute and verify the image checksum after the image is copied into Flash memory	**copy verify**

When you enter this command, the screen prompts you for the filename to verify. By default, it prompts for the last (most recent) file in Flash. Press Return to recompute the default file checksum or enter the filename of a different file at the prompt.

Clear the Contents of Nonvolatile Memory

To clear the contents of nonvolatile memory, perform the following task in EXEC mode:

Task	Command
Clear the contents of nonvolatile memory.	**write erase**

Reexecute the Configuration Commands in Nonvolatile Memory

To reexecute the configuration commands in nonvolatile memory, perform the following task in EXEC mode:

Task	Command
Reexecute the configuration commands in nonvolatile memory.	**configure memory**

Remotely Execute Commands Using rsh

You can use rsh to execute commands remotely on network servers that support the remote shell protocol. To use this command, the *.rhosts* files on the network server must include an entry that permits you to remotely execute commands on that host.

The rsh command that you issue is remotely executed from the directory of the account for the remote user that you specify through the **/user** *username* parameter.

If you do not specify the /user keyword and argument, the communication server sends a default remote username unless you override the default by configuring a remote username. As the default value of the remote username, the communication server software sends the remote username associated with the current TTY process, if that name is valid. If the TTY remote username is invalid, the communication server software uses the communication server host name as the both the remote and local usernames.

To execute a command remotely on a network server using rsh, perform the following task from EXEC mode:

Task		Command
Step 1	If you do not specify the **/user** keyword and argument in Step 2, configure the remote username to be used. This step is optional, but recommended.	**rcmd remote-username** *username*
Step 2	Enter the rsh command to be executed remotely.	**rsh** {*ip-address* | *host*} [**/user** *username*] *remote-command*

The following example shows how to execute commands remotely using rsh:

```
cs# exec
cs# rsh mysys.cisco.com /u sharon ls -a
..
.alias
.cshrc
.emacs
.exrc
.history
.login
.mailrc
.newsrc
.oldnewsrc
```

```
.rhosts
.twmrc
.xsession
jazz
cs#
```

Use Flash Memory as a TFTP Server

Flash memory can be used as a TFTP file server for other communication servers on the network. This feature allows you to boot a remote communication server with an image that resides in the Flash server memory.

In the description that follows, one communication server is referred to as the Flash server, and all other communication servers are referred to as client communication servers. Example configurations for the Flash server and client communication servers include commands as necessary.

Prerequisites

The Flash server and client communication server must be able to reach one another before the TFTP function can be implemented. Verify this connection by pinging between the Flash server and client communication server (in either direction) using the **ping** command.

An example use of the **ping** command is as follows:

```
cs# ping 131.131.101.101 <Return>
```

In this example, the IP address of 131.131.101.101 belongs to the client communication server. Connectivity is indicated by !!!!!, while ... [timed out] or [failed] indicates no connection. If the connection fails, reconfigure the interface, check the physical connection between the Flash server and client communication server, and ping again.

After you verify the connection, ensure that a TFTP-bootable image is present in Flash memory. This is the system software image the client communication server will boot. Note the name of this software image so you can verify it after the first client boot.

Note The filename used must represent a software image that is present in Flash memory. If no image resides in Flash memory, the client communication server will boot the server's ROM image as a default.

Caution For full functionality, the software residing in the Flash memory must be the same type as the ROM software installed on the client communication server. For example, if the server has X.25 software, and the client does not have X.25 software in ROM, the client will not have X.25 capabilities after booting from the server's Flash memory.

Configure the Flash Server

Perform the following task to configure the Flash server:

Task		Command
Step 1	Enter configuration mode from the terminal.	**configure terminal**
Step 2	Specify the TFTP server operation for a communication server.	**tftp-server system** *filename* [*access-list-number*]

The following example illustrates how to configure the Flash server. This example gives the filename of the software image in the Flash server and one access list (labeled 1). The access list must include the network where the client communication server resides. Thus, in the example, the network 131.131.101.0 and any client communication servers on it are permitted access to the Flash server filename *igs-bfpx.102.1*.

```
Server# configure terminal
Enter configuration commands, one per line.
Edit with DELETE, CRTL/W, and CRTL/U; end with CTRL/Z
Server# tftp-server system gs7-k.9.17 1
Server# access-list 1 permit 131.108.101.0 0.0.0.255
Ctrl-Z
Server# write memory <Return>
[ok]
Server#
```

Configure the Client Communication Server

Configure the client communication server using the **tftp-server system** command.

Perform the following task to configure the Flash server:

Task		Command
Step 1	Enter configuration mode from the terminal.	**configure terminal**
Step 2	Specify the TFTP server operation for a communication server.	**tftp-server system** *filename* [*access-list-number*]

The following example illustrates how to configure the Flash server. This example gives the filename of the software image in the Flash server and one access list (labeled 1). The access list must include the network where the client communication server resides. Thus, in the example, the network 131.108.101.0 and any client communication servers on it are permitted access to the Flash server filename *gs7-k.9.17*.

```
Server# configure terminal
Enter configuration commands, one per line.
Edit with DELETE, CRTL/W, and CRTL/U; end with CTRL/Z
Server# tftp-server system gs7-k.9.17 1
Server# access-list 1 permit 131.108.101.0 0.0.0.255
Ctrl-Z
Server# write memory <Return>
[ok]
Server#
```